TECHNIQUES, HARDWARE AND SOFTWARE FOR

ROBOTIC ASSEMBLY

A thesis presented for the Degree of Doctor of Philosophy in Mechanical Engineering at the University of Canterbury, Christchurch, New Zealand

by

Pham Duc Truong

1979

To Nga

for her continual encouragement and support.

ACKNOWLEDGEMENTS

I am grateful to my supervisor, Professor H. McCallion, for his patient guidance and lavish encouragements.

I also extend my gratitude to Professor D.C. Stevenson for his permission to use the Departmental facilities; to Drs S. Naguleswaran and K. Whybrew for their active interest in my work; to Messrs J.S. Smaill, H.J. Anink, O. Bolt, J.G. Hodge, G.R. Johnson, E.D. Retallick, and M.E. Webb for their help with the computing, electronic, and mechanical aspects of this research; to Miss J.M. Shelton and Mr T. Bird for their graphic and photographic assistance; and to Mrs P. Dowell and Mrs N. Jones for their typing of this thesis.

ABSTRACT

This thesis presents the analysis, design and development of aids for robotic assembly. The purpose of these aids is to extend the scope of low-cost industrial robots to the terminal aligning and insertion phases in the assembly of discrete components.

Following a brief survey of existing aids, several novel aids are investigated. These have been divided into two distinct categories : the 'active' category, suitable for the aligning phase and based on the conscious feedback of touch or force information arising from mechanical contact between the components during assembly, and the 'passive' category, suitable for the insertion phase and based on the implicit, yet direct, use of such information.

An in-depth treatment of the fundamental principles of active feedback technique forms a major section of the thesis. This is followed by a section on the hardware developed for implementing these techniques. The final section covers passive assembly and describes a simple and effective passive-assembly device.

For the theoretical aspects of the work, a pot-pourri of tools was borrowed from a range of disciplines such as Modern and Linear Algebra, Kinematics, Mechanics, and Numerical Analysis. The experimental aspects included the design, construction and testing of Digital Electronic hardware and Computer software for real-time computer control.

LIST OF CONTENTS

CHAPTER 1

INTRODUCTION

In general, the assembly of two or more mechanical components may conveniently be divided into four phases:

(i) pick-up phase: the components are collected from their respective storage areas (bins, magazines, pallets, etc..);

(ii) transport, or gross-positioning, phase: they are carried to some assembly station and brought into contact with each other;

(iii) aligning, or fine-positioning, phase: their positional errors are reduced and they are taken inside an insertion envelope, a spatial region defined by their geometry.

(iv) fitting, or insertion, phase: their positional errors are completely eliminated and they are driven home.

Most existing industrial robots can readily tackle the first two simple, pick-and-place phases. This study is concerned with the design and development of cost-effective devices and techniques which will extend the applicability of these robots to the more intricate aligning and fitting phases.

At the outset, in Chapter 2, the general field of assembly mechanisation will be surveyed in order to provide a background for subsequent work. It will be seen that current assembly systems can be grouped according to whether or not they employ feedback. In non-feedback or open-loop assembly systems, all the four assembly phases listed above are implemented by structuring the assembly environment and observing high dimensional and positional accuracy. These systems are efficient, but also expensive and inflexible, and therefore suitable only for mass-production situations.

On the other hand, feedback or closed-loop assembly systems show real potentialities for batch production. Frequently, in these systems programmable robots guided by sophisticated visual and tactile sensors are employed to execute phases (i), (iii) and (iv). As these systems can perceive and interact with their environments, they require a much lower degree of environmental structuring and achieve a much higher degree of flexibility than do open-loop systems. In the present state of the art, however, most closed-loop assembly systems are still relatively slow, expensive and capable of performing only simple and artificial tasks.

From the survey, it will be apparent that the viability of batch-oriented robotic assembly hinges on the development of improved feedback techniques. Therefore, in Chapter 3, the study will focus on the problem of feedback. Initially, the threads left in the survey will be picked up with a comparative examination of current sensing techniques, this time from a more abstract viewpoint. The concept of sensor and error spaces will be introduced, and sensing regarded as a relation between the two spaces. In this light, visual sensing will approximate to a functional relation, while unilateral contact sensing will show up as a one-to-many relation. Unilateral contact (tactile) sensing, the only form of contact sensing developed to date, involves obtaining feedback information from a single array of sensors fixed to either a robot or a work table. It will be shown that contact sensing can theoretically be made functional if bilateral information is extracted from two sensor arrays, one fixed to the robot and the other to the work table. The Chapter will close with the description of a family of feedback techniques based on this bilateral principle.

Work on bilateral feedback will continue in Chapter 4 where detailed examples will be provided to illustrate both the theoretical and practical facets of selected bilateral techniques. It will be shown that in order to apply these techniques, either the robot or the work table must be capable

of making small and accurate movements during the detection and correction of positioning errors.

Because in most assembly operations the robot has to perform gross material-handling tasks involving large and swift movements, it seems logical that the finer corrective steps be carried out by the work table. To achieve high positioning accuracy as well as reasonable operating speeds, the work table must be rigid, but also light and manoeuvrable. Chapter 5 will be devoted to the analysis of an original computer-controlled work table which possesses all the required characteristics. It is envisaged that, by using this machine in conjunction with the bilateral techniques proposed in Chapters 3 and 4, even a modestly accurate robot will be able to perform the aligning phase without difficulty.

Finally, in Chapter 6, a simple device will be described which, when coupled to such a robot, will help it succeed also in the insertion phase. The device has been proven on a Unimate 2000B Industrial Robot: the Unimate, which under normal circumstances can merely position its hand to ± 1.2 mm repeatability, has become capable of consistently inserting pistons into cylinders with diametral clearances as fine as 0.02 mm.

CHAPTER 2

A SURVEY OF MECHANISED ASSEMBLY

The term 'mechanised assembly' is used here in its broad sense to cover the production, mainly by machines, of objects from an aggregation of smaller objects.

There are two approaches to mechanised assembly. The first approach is to use special-purpose fixed or programmable machines with high accuracy and rigidity and no sensory feedback; the second approach involves flexible robots with sensors, interactive skills and decision-making abilities.

This chapter outlines assembly systems representative of these open-loop and closed-loop approaches and exposes their underlying principles and techniques.

2.1 OPEN LOOP ASSEMBLY SYSTEMS

2.1.1 Description

A fixed open-loop assembly system [1] [2] is, in general, made up of several work stations. Each station is equipped with workheads and part feeders and is joined to the next station by a conveyor. The assembly operation is shared between the work stations, the work being built up on a work carrier and transferred from station to station on the conveyor. At each station, the work carrier is firmly and accurately located by jigs and fixtures while workheads perform part placing and fastening or inspection tasks. Workheads may be supplied with either magazined or individual parts from hopper-feeding and orienting devices.

In addition to all the basic components of a fixed system, a

programmable system [3] possesses a programmable controller which synchronizes and co-ordinates the various work stations. Therefore, the system's operating sequence may be readily changed by changing the programs stored in the controller's memory.

To date, many fixed and programmable open-loop assembly systems have been developed and successfully applied to large and medium-scale production. Fixed automatic transfer lines have been dedicated to the repetitive assembly of one type of item, such as lamps [5], microphones [5], gear boxes [5], and water pumps [6]. Programmable assembly machines, on the other hand, have been employed for producing groups of similar items, for example, a family of post office relays [7], vehicle locks [8], or printed-circuit boards [9].

Finally, programmable assembly systems making extensive use of current industrial robots have also been investigated. The most impressive effort in this direction has been demonstrated by Kawasaki Heavy Industries whose prototype engine assembly shop features ten Kawasaki-Unimate robots and a minicomputer for controlling them [10]. There are five work stations, each manned by two robots and equipped with a work platform, a hydraulic press and the traditional array of jigs, fixtures and feeding and orienting devices. In this set up, mechanisation is only partial as humans have to be employed in two of the twelve stages forming the complete assembly operation.

2.1.2 Principles

Since an open-loop system, by definition, has no feedback mechanism to help it cope with uncertainties, it must work in a structured environment in which these uncertainties are minimised. For assembly systems, uncertainties may exist in the dimensions of parts, their alignment and orientation, and the positioning of the workheads. The principle of open-loop assembly is then that of observing high dimensional and

positional accuracy. This means (1) parts must have tight tolerances and must be accurately located relative to workheads, (2) workheads positioning must be precisely repeatable, and (3) the assembly machine must be highly rigid.

2.1.3 Discussion

The determinate nature of a structured assembly environment is conducive to high output rates but its stringent requirements on accuracy have created a heavy reliance on expensive jigs and fixtures. In consequence, an open-loop assembly system gets so costly that it can only be used for large-scale production in which a million identical assemblies, or more, are required per year, over a period of several years [3].

The incorporation of programmable controllers which facilitate minor product changes has extended the scope of open-loop machines to the domain of medium-scale production. However, since without feedback accurate jigging is still mandatory, even programmable machines do not seem economical for small-batch assembly [11].

Attempts have been made to reduce the environmental structuring in open-loop systems (and hence their costs) by fitting them with special assembly aids and thus enabling them to tolerate some degrees of uncertainty in their environment. For example, parts need not be located exactly, if chamfers are present and compliant workheads are used [12] [13] [14] [15]. Dynamic aids involving random vibration [16], ordered search [12], or air flow [17] [18] [19] have also been reported, which can correct small initial misalignments between parts and workheads.

However, assembly aids which are, of necessity, job-oriented cannot increase the versatility of an assembly system: a special device for assembling cylindrical pegs and holes is in general useless with square

pegs and holes. To be truly versatile, then, a system must have sensors which empower it to perceive its environment and cope with any arbitrary changes therein.

2.2 CLOSED-LOOP ASSEMBLY SYSTEMS

2.2.1 Description

In general, closed-loop assembly systems are physically very similar to the programmable machines described in the previous section. That is, a closed-loop system also employs a programmable controller (usually a minicomputer) and a number of parts feeding, orienting, conveying and placing devices. The distinctive feature of a closed-loop system is that, here, the controller can modify the system's behaviour by making real-time decisions based on the environmental information gathered by the system's feedback sense.

A closed-loop assembly system can be classified according to the types of sensing it uses. There are currently three types of systems: (1) those with contact sensing, (2) those with non-contact sensing, and (3) those with both types of sensing.

2.2.1.1 Assembly systems with contact sensing

Contact sensing involves the detection of force, pressure, or binary contact at the interface between a system and its environment [20] [21].

Binary contact (or touch) sensing is the simplest of all contact sensing techniques. Mechanical microswitches fitted to the hand of a manipulator are usually the cheapest and most reliable touch sensors, although magnetic switches [22] [23] and even analogue pressure and force transducers [24] [25] [26] [27] have also been applied.

Pressure sensing, on the other hand, can be easily implemented with binary switches [28] [29] [30]. Proportional transducers found

in existing robot systems include piezoelectric crystals [31] which emit electric currents when deformed, and graphite cells [32] or conducting polymeric fibres [32] [33] which change in resistance under pressure. An original transducer has also been reported, in which the amount of light received by a phototransistor varies directly with the applied pressure [34]. Pressure transducers are usually arranged into a matrix and embedded in some elastic and insulating medium where they act like nerve endings in the human skin.

Force (and moment) sensing can be practised at the fingers [30], the wrist [25] [26] [35] [36] or other joints of a manipulator [24] [37] [38], or can be incorporated into a sensing pedestal separate from the manipulator [35] [39]. Rugged and inexpensive strain gauges are perhaps the most popular devices for finger, wrist and pedestal force sensing, but combinations of springs and displacement transducers can also be used for the same purpose [26] [40] [41]. The feedback of joint forces and moments generally involves measuring motor currents or differential hydraulic pressures on the appropriate joint actuators [42]. Although it is convenient, joint force sensing is an inaccurate method of determining the forces on a manipulator's hand, because of the variability of the manipulator's inertia and the non-uniform friction in the individual joints [20].

In spite of its apparent simplicity, contact sensing has not been widely applied. So far, the only commercial assembly systems equipped with contact sensors are the Hitachi's Hi-T-Hand machines for cylindrical peg-hole insertion tasks and the Olivetti's SIGMA machines for more general assembly operations.

The Hi-T-Hand Expert-1 model [25], used for the assembly of pistons into cylinders with 20-micron clearances, has a simple pin-board sequential controller and two pick-and-place robots. In a typical

assembly cycle, the main robot picks up a piston, and the auxiliary robot, a cylinder. Both components are then brought together, but the positioning accuracy of the robots is insufficient for direct mating to take place. The main robot corrects the initial piston-cylinder misalignment by moving the piston in the direction in which it tips when it contacts the cylinder's mouth. This direction is detected with strain gauges mounted on the compliant wrist of the main robot. When the misalignment becomes sufficiently small, the springs on the main robot's wrist automatically push the piston into the cylinder, and the insertion phase begins. During this phase, the main robot continuously makes small hand displacements based on the sensor's feedback in order to avoid jamming the piston against the cylinder's wall.

The more recently developed Hi-T-Hand Expert-5 [23] can assemble end brackets of electric motors into ball bearings with clearances around 5 microns. It is also sequentially controlled and is equipped with only a simple binary switch for contact sensing. With the bearings held firmly in a horizontal plane, the Expert-5 performs the assembly by pushing the brackets down until the switch is activated, which indicates the presence of an obstacle, and then moving them horizontally in a spiral search pattern to dodge the obstacle. The machine may need to alternate several times between pushing and spiralling to complete a successful assembly cycle.

SIGMA machines have been employed for a variety of assembly operations, such as the production of typewriter sub-assemblies and the insertion of IC chips into printed-circuit boards [26] [43]. The basic machine has two computer-controlled arms which can travel along overhead guide rails. Each arm is equipped with a flexibly-mounted hand and force-displacement transducers for measuring the forces applied

to the hand and the location of the hand relative to the arm. This feedback information helps the system to protect itself and the components it assembles, check the progress of an assembly operation, or compensate for any small positional discrepancies during the operation. SIGMA can continuously harness feedback for implementing search manoeuvres similar to those in the Hi-T-Hands. In general, however, it employs feedback in a binary mode for making conditional tests ("are parts present or absent?", "are dimensions correct or abnormal?" etc.) and branching to the appropriate sections in its control program.

Contact sensing assembly has also been investigated by a number of research institutions including the MIT Draper Laboratory [44] [45], the Stanford Research Institute [46] [47], and the University of Canterbury [48] [49]. So far, the emphasis appears to have been in studying the fundamentals of the assembly process, deriving useful contact-sensing strategies, and investigating basic part-mating operations (such as threading a nut onto a bolt or inserting a peg into a hole). One of the few experimental contact-sensing systems used for more comprehensive assembly work is the Little Robot System of the MIT Artificial Intelligence Laboratory. This system has a computer-controlled arm with Linear-Variable Differential Transformers (LVDTs) mounted at the wrist for force and torque measurements. It can assemble an 8-part bearing complex, a task which involves screwing a nut onto a bolt, inserting pegs into holes with small clearances, and picking up thin washers from a flat surface [50]. A feature of the system is its extensive application of force feedback in implementing active accommodation. Active accommodation consists of making small displacements to comply with external constraints and is, in general, only suitable for correcting small positioning errors. For example, to house a partially inserted peg, the arm is programmed to move the

peg so as to nullify the lateral forces and tilting moments acting on it, which usually means moving it in the direction of the sensed forces and moments.

2.2.1.2 Assembly systems with non-contact sensing

Non-contact sensing enables a system to identify and locate objects, measure their distances to a datum, or simply detect their presence in, or absence from, a scene, without making mechanical contact with its environment.

The non-contact detection of objects can be implemented with proximity sensors. Various types of proximity sensors are commercially available. These can be grouped into air-jet devices for the short range detection of solid objects [51] [52], permanent-magnet devices for the short-range detection of ferrous objects; inductive devices for the short-range detection of metallic objects; capacitive devices for the short or medium range detection of any object; and transmitter-receiver type devices using ultrasonic or light beams for the short, medium, or long-range detection of any object [53].

The distance of a point to a datum can be measured by range finders. There are three major range-finding schemes [32]: the first is based on transmitting laser or sound pulses and measuring the arrival time of the reflected signals [30] [54] [55], the second, on sending laser or sound waves and determining the phase shifts of the reflected signals [56] [57], and the third, on transmitting continuous frequency-modulated waves and measuring the instantaneous differences between the frequencies of the transmitted and received signals [58].

The identity and location of objects can be determined visually with electro-optical imaging sensors [20]. The simplest of these

devices is the linear diode array which consists of between 16 to 1872 photodiodes arranged in a straight line [59]. An orthogonal pair of arrays used in conjunction with a moving scene (e.g. objects on a conveyor) can register three-dimensional pictures of the scene [60]. The taking of two-dimensional pictures is, in general, performed directly with standard television cameras (vidicon, silicon-vidicon, plumbicon, etc.) [61] or with solid-state diode-matrix cameras (also known as area-array cameras) [59]. These cameras operate on a sequential raster-scan mode, a raster comprising up to approximately 415 x 625 picture elements (pixels) in a standard TV camera and between 32 x 32 and 320 x 512 pixels in an area-array camera. Another type of camera is the image-dissector camera which is capable of measuring light intensities at arbitrarily selected points. Image dissectors can achieve resolutions higher than 1000 x 1000 pixels [62].

Assembly systems which employ non-contact sensing as their sole guidance method have been mostly confined to artificial research environments. In assembly research, it is the third category of non-contact sensing, namely, visual imaging, which has received the most attention. To illustrate visual feedback assembly and its potential, four experimental systems are now described. They are (1) The Nottingham SIRCH general-purpose assembly machine, (2) the Mitsubishi motor-brush insertion robot, (3) the General Motors' wheel mounting system, and (4) the Hitachi Central Research Laboratory's intelligent robot.

The SIRCH assembly machine [63] [64] consists of a manipulator with three different hands, a TV camera and a small computer. The manipulator's hands and the camera's 'eye' are mounted on a common turret, the mechanical arrangement being equivalent to one with the eye incorporated in the hands. SIRCH employs heuristic visual

recognition methods. First, it obtains an electrical image of a scene by sampling the brightness of the scene at a fixed number of points, converting the brightness values into binary data and storing them in the computer memory. Second, it spatially differentiates the stored image and reduces it to an outline 'diagram' in which edges are represented by 1's and everything else by 0's. Third, it chains the edges into closed contours and calculates the areas, perimeters, and centroid co-ordinates of the objects enclosed by the contours. Fourth, with these statistics, it tentatively identifies the object to be picked up, moves the eye-hand assembly to a position above the object's centroid and 'zooms in' on the object. Fifth, it takes a close-up picture of the object and repeats the outlining and edge-tracing algorithms to obtain more accurate measurements of the object's features including its area, perimeter, and centroid co-ordinates, the number of holes in the object, the sizes and locations of the holes, and the polar co-ordinates of points on the object's periphery. Finally, using these new feature measurements, it decides whether the object under scrutiny is the desired object. If so, it computes the co-ordinates of the point from which the object is to be picked up and makes successive moves to reach the pick-up location where it selects the appropriate hand and acquires the object. Otherwise the machine backs off and chooses an alternative object for examination.

The Mitsubishi assembly system [65], used for inserting brushes into DC motor sub-assemblies, comprises a brush-feeding unit, a brush-handling robot, a TV camera, a brush-inserting machine and a minicomputer. The system also adopts the eye-in-the-hand approach, with the camera mounted directly between the two fingers in the robot's hand. The main tasks of the robot are to place a brush holder in the inserting machine, measure the position and orientation of a brush as

it is presented by the feeding unit, pick up the brush accurately, place it in the inserting machine, and remove the brush holder when four brushes have been inserted. Position measurements are made by displacing the camera and analysing the parallax which results from viewing the scene (a brush) from two separate stations. Orientation measurements are made by rotating the camera in a prescribed sequence until the observed image aligns with an internally stored image.

The General Motors' experimental system for mounting wheels onto automobile hubs [66] has a manipulator, a hub supporter, a TV camera, and a computer. At the beginning of an assembly operation, the experimenter roughly locates a hub on the supporter and slips a wheel into the manipulator's hand. The camera then takes a picture of the hub and processes the picture to determine the hub's centre and orientation. While the processing takes place, the manipulator brings the wheel into the field of view of the camera. Next, the picture of the wheel is taken and its centre and orientation are found. The displacements between the hub's studs and the wheel's corresponding stud holes are calculated and the results fed to the manipulator which then makes appropriate movements to mount the wheel onto the hub. The pictures are always taken in a plane perpendicular to the axis of the hub (or the wheel) so that the input patterns always assume flat, symmetrical, circular shapes. Owing to this circular symmetry, the algorithms for finding the hub's (or wheel's) centre and orientation can make use of efficient signal-processing techniques (peak detection, thresholding, and convolution filtering) instead of the tedious visual recognition procedures described above.

The Hitachi Central Research Laboratory's intelligent assembly robot [67] is equipped with a mechanical hand, two TV cameras and a computer. The robot, which exhibits both visual pattern recognition

and imaginative reasoning abilities, has been programmed to assemble three-dimensional objects from plan drawings. With one of the cameras, it examines the drawings then deduces the three-dimensional structures of the objects, breaks the objects into their component parts and formulates assembly strategies. With the other camera, it looks at the real parts placed randomly on a table, identifies the required parts, and makes decisions on how to manipulate these parts to produce the objects in the drawings. Object recognition is done by spatially differentiating the camera pictures to obtain sharp outlines, fitting straight segments onto the outlines and extracting the geometric features of the resulting patterns. Drawings understanding is achieved by having a pre-programmed set of logic rules which enable the robot to 'reason' like humans do when they interpret engineering drawings.

2.2.1.3 Integrated assembly systems

Integrated assembly systems are assembly systems with both contact and non-contact sensing. The sensors found in integrated systems are, in essence, the same as those already described in conjunction with contact-sensing and non-contact-sensing assembly systems. In general, non-contact sensors (TV cameras, etc.) are used for global sensing with coarse resolution while contact sensors (binary touch, pressure, or force transducers) are for local sensing and finer resolution.

Although to date a number of integrated assembly systems have been developed, only one system is believed to have reached commercial status. This system, developed by Hitachi, is equipped with a TV camera and touch sensors and is used to tighten bolts in moulds for concrete piles during the casting and curing process [68]. As the moulds move along the production line, the bolts are recognised by the camera which also determines their approximate positions. A robot arm then moves in,

locates the bolts accurately using its touch sensors, and tightens them with an impact wrench.

Other integrated assembly systems include the experimental systems built at the University of Edinburgh, IBM, Stanford University and Hitachi Central Research Laboratory.

The Edinburgh versatile assembly equipment consists of a movable table, a mechanical hand with force sensors, and two TV cameras, all connected to a supervisory computer [37]. This system has been taught to put rings onto pegs and construct simple devices such as a toy car and a toy ship [69]. Teaching is by showing the parts to be assembled to one of the cameras and 'leading' the hand through the layout and assembly motions. During this process, the machine creates internal models (or hierarchical descriptions) of the parts and memorizes the assembly sequence. The parts are subsequently dumped in a heap on the table. The machine breaks the heap, recognizes each part by matching its structure with one of the internal models and lays it out in its standard position and orientation. The machine then assembles the parts, helped by force feedback in the final fitting stage. Force feedback is used to implement searches as in the Hitachi and Olivetti systems already described.

The IBM's experimental assembly set-up comprises a robot with force, touch, and proximity sensors, a work table, and a supervisory computer system [70]. There are also provisions for equipping the robot with a comprehensive array of tactile, visual and acoustic range-imaging devices [30]. The set-up is intended for the class of assemblies which measure less than 300 mm cube and contain up to 50 parts each weighing between 1 g and 500 g. Programs have been written to stack toy blocks, assemble a toy train, and construct a complex

20-part typewriter sub-assembly that includes two screws and a spring. Again, touch and force feedback is used primarily to implement blind searches, although novel work has also been done on using force feedback for directly measuring the position and orientation of the components to be assembled [72].

At Stanford University, a programmable assembly system has been developed which incorporates a robot (two, in some experiments) with touch sensors and joint force transducers, a TV camera, a work station with simple fixtures and tools, and a pair of control computers [27] [38]. The system has been used for a variety of tasks, including the assembly of a pencil sharpener, a door hinge and an automobile water pump. The judicious integration of the different sensory modalities is well illustrated by the water pump assembly operation [39]. A water pump consists of a base, a gasket, a top (including a rotor), and screws. First, with the TV camera, the robot roughly locates the base and then grasps it accurately, guided by touch sensors mounted between the robot's fingers. Next, the robot places the base in its standard position and orientation against fixed aligning blocks. Using force feedback, the robot searches for two of the screw holes in the base and inserts guide pins into them. It then places the gasket on the base and visually inspects the gasket's position. Finally, it mounts the pump top onto the base, withdraws the guide pins, and secures the top to the base by means of the screws. Force feedback enables the robot to control the torque it applies when screwing and to check that the rotor can turn freely after the screws have been inserted.

The integrated assembly equipment at Hitachi Central Research Laboratory comprises two co-ordinated robot arms (the heavy POWER arm and the light SENSOR arm) fitted with an impressive battery of 30

touch, force and pressure sensors, seven stationary TV cameras, a mobile camera (held in the hand of SENSOR arm), an image processor, and two control computers [73] [74]. This equipment has been programmed to assemble a vacuum cleaner consisting of a filter unit, a motor unit and a dust case. The assembly operation has been divided into six stages. First, from a heap of identical filter units, that unit which can be most easily picked is recognised by the visual system using one of the stationary overhead (vertical) cameras which also determines its macroscopic position. SENSOR arm then approaches this position and, with its movable 'eye', precisely locates the filter's rim. This information enables the arm to pick up the filter unit by the rim, the final pick-up steps being guided by touch sensors in the arm's extremities. Second, with the assistance of POWER arm, SENSOR arm finds a stable and convenient location in the centre of the filter unit and regrasps it there. This unusual handling operation involves touch sensing in alternate arms. Third, POWER arm picks up the dust case and carries it to the assembly area. There, POWER arm holds the case, and SENSOR arm inserts the filter unit into it. During insertion, information on the state of contact between these two components is relayed to the active SENSOR arm by means of contact sensors in the now-passive POWER arm. Fourth, POWER arm picks up the motor unit, an object with a shiny plastic cover. The vision system, with adaptive light-intensity thresholding abilities, enables the object's approximate location to be determined. Again, touch sensors in the tip of the arm are subsequently used for guiding the precise grasping of the object. Fifth, POWER arm mounts the firmly grasped motor unit onto the dust case. Three stationary cameras (2 horizontal and 1 vertical) survey this operation, each camera measuring the linear positioning errors in the horizontal x and y axes and the angular misalignment about the

vertical z axis, respectively. Sixth, SENSOR arm and then POWER arm, each presses an over-centre catch to lock the motor unit to the dust case. This locking operation is monitored by force sensors in the wrists of the arms.

2.2.2 Principles.

Since a closed-loop assembly machine is endowed with sensors for perceiving its environment, it no longer needs to operate by 'dead reckoning', but is capable of working in unstructured situations where there may be variations in the dimensions of the components to be assembled and uncertainties in their initial alignment and orientation.

Unlike in open-loop assembly machines which have to maintain high dimensional and positional accuracy by means of structural rigidity and jigs and fixtures, the principle here is gradually to achieve the required degree of functional accuracy through intelligent utilization of sensory information. The required functional accuracy is determined by the geometrical properties of the components being assembled. For example, if the components are a peg and a hole, the functional accuracy to be achieved by a positioning system is directly related to the diameters of the peg and the hole.

In general, contact-sensing assembly machines achieve functional accuracy through trial and error and heuristic methods. Consider the common task of placing a peg (a shaft or a piston) into a hole (the bore of a bearing or a cylinder).

Initially, the sensors (touch, force, or pressure sensors) may show null readings, which indicates that most probably the peg is located outside the hole. This can be confirmed if the sensor outputs remain null after the peg has been displaced by small amounts in various directions relative to the hole. As the peg is brought nearer to the

hole, the sensors will eventually register contact. If it can then be assumed that the peg has not hit a foreign obstacle, some of the possible states of the peg are (i) it is touching the plane of the hole entrance at one point, (ii) it is standing squarely on this plane, next to the entrance, (iii) it is contacting the entrance at two points, (iv) it is on the chamfer, (v) it is jammed against the hole, and so on.

To determine that state with the maximum likelihood of being the actual state, various tests based on contact sensing must be performed. A logical test, applicable when force sensors are used, may be to feel the resistance of the peg to rocking while still in contact with the component containing the hole. If no resistance is felt as the peg is rocked about any arbitrary axes, state (i) is highly probable. If the peg offers equal resistance to rocking about all axes, state (ii) is a good candidate. On the other hand, if the peg shows particular susceptibility to rocking about a single axis, either state (iii), (iv) or (v) may be true, and so on.

Subsequent motions to be imposed on the peg depend on the outcome of this and similar tests. For instance, if it has been decided that the peg is currently standing next to the hole entrance (whose location relative to the peg is still unknown), the peg may be set moving in a spiral to search for this entrance. Searching halts when a change in the contact state is detected. More tests are then performed and appropriate motions imposed until, after exhaustive attempts, the final state is reached when the peg is properly seated inside the hole.

From the above example emerges a pattern common to existing contact sensing assembly machines: due to the ambiguity of contact information, they cannot perceive with absolute certainty the true current state of an assembly operation (which, in the example, is represented by the actual misalignment between the peg and the hole), and, hence,

are restricted to postulating likely states, verifying their postulates, and iterating 'blindly' towards the goal state.

On the contrary, assembly machines with non-contact sensing are, in general, capable of 'seeing' the current state of an operation and, thus, of achieving functional accuracy by deterministic methods. In the case of peg-hole assembly, for example, they will measure (using vision or otherwise) the misalignment between the peg and the hole and then position the peg accordingly. All measurements are based on physical or geometrical principles. If initially the peg and the block containing the hole lie on known supporting surfaces (say, the top of a work table) and if a TV camera is used for sensing, then the 'support hypothesis' method [75] can be applied to enable the camera to measure distances and angles from a fixed viewing station. If no a-priori information is given on the initial peg-hole location, then 'parallax' (using a moving camera) or 'stereoscopic vision' (using 2 or more cameras) methods based on the triangulation principle [75] may be necessary to yield three-dimensional measurements from the observed two-dimensional images.

Naturally, integrated assembly machines, which have both contact and non-contact sensing, resort to both trial-and-error and deterministic methods to achieve functional accuracy. Again, in assembling a peg into a hole, they may at first use visual sensors for measuring the gross peg-hole misalignment and then touch sensors for guiding the peg into the hole when vision is occluded or the visual sensors have reached the limits of their resolving power.

2.2.3 Discussion

Batch production is characterised by frequent product changes. To be suitable for batch production an assembly machine must be able to cope with these changes without involving major hardware modifications

and machine-downtimes. In addition, its cost must be reasonably low, and its speed, at least commensurate with that of human workers.

In the present state of the art, it is generally impossible to meet these conflicting requirements on flexibility, cost, and efficiency, all at once. For example, consider contact-sensing assembly. It would be relatively easy to construct an inexpensive machine with off-the-shelf sensors, floppy robot arms, and a simple programmable controller. The machine could be made quite flexible by minimising its reliance on complex peripherals (jigs, fixtures, and feeding and orienting devices). However, this reduction in environmental structuring would be accompanied by a proportional increase in the ambiguity of contact information. In consequence, the required search strategies would be very elaborate and the machine, extremely slow. To achieve realistic speeds, existing contact-sensing assembly machines have had to maintain a considerable degree of environmental structuring, thus sacrificing flexibility and increasing cost.

On the other hand, non-contact-sensing assembly machines are inherently flexible by virtue of their high level of 'consciousness'. As seen previously, even machines at the lower end of the flexibility spectrum can recognise randomly placed objects, determine their locations, pick them up, and assemble them according to pre-taught sequences. In the machines developed to date, the key operation, object recognition, is usually performed by processing large matrices of visual data (these matrices often consist of between 50,000 and 100,000 elements). This complex operation is time-consuming and necessitates the use of either an image processor or a general-purpose computer with sophisticated software. It is obvious, then, that the two major drawbacks of these machines are their slow response and relatively high cost.

At present, non-contact-sensing assembly machines also suffer from low resolving power. In visual systems, this is due to non-linearities in the cameras, their susceptibility to noise from the environment, and digitising errors committed in the processing of pictures. The resolution obtained with most machines is seldom better than a few millimetres and, therefore, unacceptable for the majority of real assembly tasks. This explains why non-contact-sensing assembly is still restricted to laboratory situations where only artificial tasks are performed.

Finally, there are integrated assembly machines, in which the inherent flexibility of non-contact sensing is augmented by the naturally high sensitivity of contact sensing. As a result of this logical association, these 'intelligent' machines are capable of performing genuine industrial tasks involving the type of tolerances once negotiable by humans only. Although low speed is again a problem with these machines, it will certainly be overcome when more work is furthered into developing and refining the current techniques of visual recognition and tactile sensing. What seems at present insuperable is the problem of complexity associated with the high order of intelligence of these machines. Naturally, complexity means high cost, and this is the ultimate barrier to the widespread application of integrated assembly.

2.3 SUMMARY

The vast domain of mechanised assembly has been surveyed and partitioned into two distinct areas known respectively as open-loop and closed-loop assembly.

Open-loop assembly has been briefly analysed and found to depend on maintaining absolute accuracy throughout the assembly process by means of well-built hardware systems and highly structured environments.

Consequently, open-loop assembly is efficient but inflexible, very expensive, and applicable only to medium-scale and mass production.

Closed-loop assembly, a potential candidate for small-batch production, has been studied in more detail. Representative closed-loop machines have been described and grouped according to the sensory modality they employ - contact sensing, non-contact sensing, or both. Currently, each group has its strengths and weaknesses although, in theory, all can tolerate unstructured environments by systematically eliminating uncertainties through the use of sensory information. Practical contact-sensing assembly machines are fast, sensitive, but relatively inflexible. Non-contact sensing machines, on the other hand, are much more adaptable, but also complex, slow and inaccurate. Integrated machines, equipped with both contact and non-contact sensors, can achieve accuracy and flexibility comparable to human standards. These machines, however, are still troubled by the problems of low speeds and unrealistic costs.

CHAPTER 3

ON FEEDBACK TECHNIQUES

The viability of batch-oriented robotic assembly hinges on the feedback techniques employed. Therefore, in this chapter, we shall focus our attention onto the problem of feedback.

The chapter begins with a brief comparative examination of current feedback techniques from an abstract viewpoint. The concept of sensor and error spaces is introduced and sensing is regarded as a relation between the two spaces. This preamble into the abstract will lead us to the formulation of a new feedback principle. A discussion of a family of feedback techniques based on this principle concludes the chapter.

3.1 SENSOR AND ERROR SPACES

The sensor space of an assembly machine is defined as the set of all the possible states of its sensors. For example, if a machine employs a binary switch as its only sensing device, its sensor state is finite and composed of two elements. If the machine has six binary switches, the number of elements increases to 2^6, or 64. On the other hand, if the sensors are analogue devices (strain gauges, photo-transistors, etc.), then the sensor space is theoretically an infinite set. Each element of a sensor space is a vector whose components characterise the states of the individual sensors. The number of components in a vector can range between 1 (as in the single-switch machine) and several thousand (as in a visual robot).

The error space during the assembly of two rigid bodies is defined as the set of all the possible misalignments between them. The error space is theoretically infinite. In general, each of its elements is a 6-dimensional vector specifying the relative position and

orientation of the two mating bodies.

The assembly of two bodies may be regarded as a problem of matching their respective coordinate systems, and sensing, a case of deducing the systems' misalignments from the states of the sensors involved. That is, sensing (or more correctly, perception) is a relation from sensor space to error space (Fig. 3.1a).

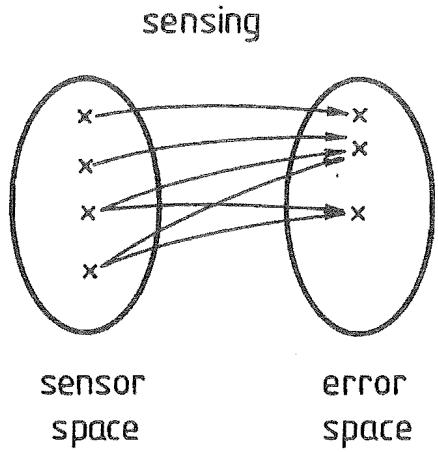
A general observation can be made at this point. Ideally, sensing should be an unambiguous procedure, which means that the relation from sensor space to error space should be functional (or that to each element in sensor space there should correspond one and only one element in error space) (Fig. 3.1b). As an error-space element is defined by six independent quantities, a necessary condition for functionality is that each element in sensor space has at least six independent co-ordinates.

3.2 CURRENT SENSING TECHNIQUES

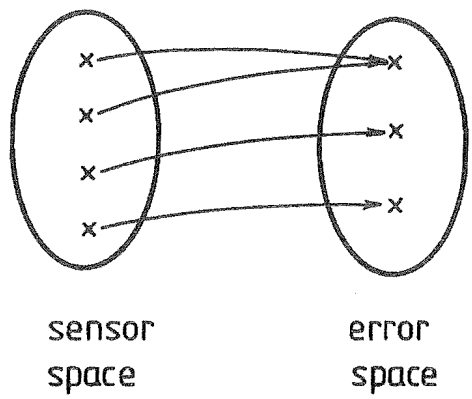
Current sensing techniques have already been described in Chapter 2, in connection with closed-loop assembly systems. In this section, we shall attempt to provide somewhat more formal explanations for their inherent properties. The simple but classical example of peg-hole assembly will sometimes be used to illustrate abstract concepts although these can apply to more general situations. The discussion will follow the pattern of Chapter 2 and will proceed from the simple contact-sensing techniques to the more complex non-contact and integrated techniques.

3.2.1 Contact sensing

In all of the assembly systems that we have reviewed, contact sensing may be described as unilateral. That is, it involves extracting information from a single array of sensors mounted either on a robot or on a work table (even where two arrays are employed, as in the Hitachi



(a) sensing as a relation



(b) sensing as a <u>functional</u> relation

FIG 3·1 SENSOR AND ERROR SPACES

integrated assembly system [73], contact information is still derived from a single array during any given phase of an assembly task).

If, as in some contact-sensing machines [23], the sensor array consists of only one binary switch, then clearly the necessary condition for functionality stated in Section 3.1 is not satisfied. Consequently, contact sensing with a binary switch is a one-to-many relation from sensor space to error space (Fig. 3.2a). In more practical terms, a robot fitted with a simple binary switch cannot uniquely determine the relative location between a part fixed to a work table and another held in its hand, unless it uses the switch for performing searches as a blind man does with his cane (like the man, the robot then needs to combine its touch sense with a sense of position or proprioceptive sense).

In most contact-sensing robots developed to date, the sensor array is a six-degree-of-freedom force (and moment) transducer. A sensor-space vector then consists of six components. However, as we shall now demonstrate, it still cannot be mapped unequivocally to a counterpart in error space.

First, note that the outputs of a force transducer are continuous, but in some applications they are used simply to signal binary contact events along the different axes of the transducer [26] [38] [69]. For example, while assembling a peg into a hole, a robot often is concerned - not with the true contact force and moment vectors on the peg - but with whether they exceed a given threshold along, say, the transducer's axis corresponding to the peg's axis. In such a case, a sensor-space element is effectively a vector with six binary components whose states characterise the states of the contact forces in the respective transducer axes. The sensor space then consists of a finite number of elements. On the other hand, the error space is infinite and each of its elements is the 'image' of at least one element in sensor space. Since a finite space

cannot be mapped onto an infinite space, binary-vector contact sensing is clearly not a functional relation (Fig. 3.2b). Again, this means the robot must resort to blind search as it is unable to pin point the relative location of mating parts with the information derived from a single contact between them.

Finally, in the case of analogue-vector contact sensing where the sensor space is infinite because the reaction forces and moments on mating parts are recorded as true vector quantities, it can be shown that each sensor-space element is equivalent in geometric information content to a vector with only five components at the most. The geometric information content of a sensor-space vector is that part of the vector which can provide clues about the error space. (Geometric information is to be distinguished from stress, deformation, or other types of information picked up by a sensor array).

To show that a sensor-space vector consisting of a reaction force $\underset{\sim}{F}$ (with 3 components) and a reaction moment $\underset{\sim}{M}$ (with another 3 components) contains no more than 5 pieces of geometric information, we shall invoke a theorem by Poinsot on the reduction of systems of forces and moments [76]. In essence, the theorem states that, except when $\underset{\sim}{F}$ is null, the intrinsic resultant of $\underset{\sim}{F}$ and $\underset{\sim}{M}$ is a wrench which comprises a force acting along a straight line and a couple, both parallel to $\underset{\sim}{F}$. Clearly, only the line of action of the force (or the axis of the wrench) and, perhaps, the ratio of the magnitudes of the force and the couple (or the pitch of the wrench) are of potential geometric relevance in the determination of the location of mating parts (whereas the absolute magnitudes of the force and the couple would be important in the determination of stresses and deformations in the parts). Since four parameters are required to define the axis of a wrench and one parameter, to define its pitch, it follows that when $\underset{\sim}{F}$ is non-zero, there are five available

independent items of geometric information. Of course, if $\underset{\sim}{F}$ but not $\underset{\sim}{M}$ is null, we have a pure couple and hence only three such information items.

Thus, in practice the number of geometrically significant components in a force-moment vector is always less than six. Again, by referring to the functionality condition stated in section 3.1, we can conclude that analogue-vector contact sensing, too, is a one-to-many relation between sensor and error spaces.

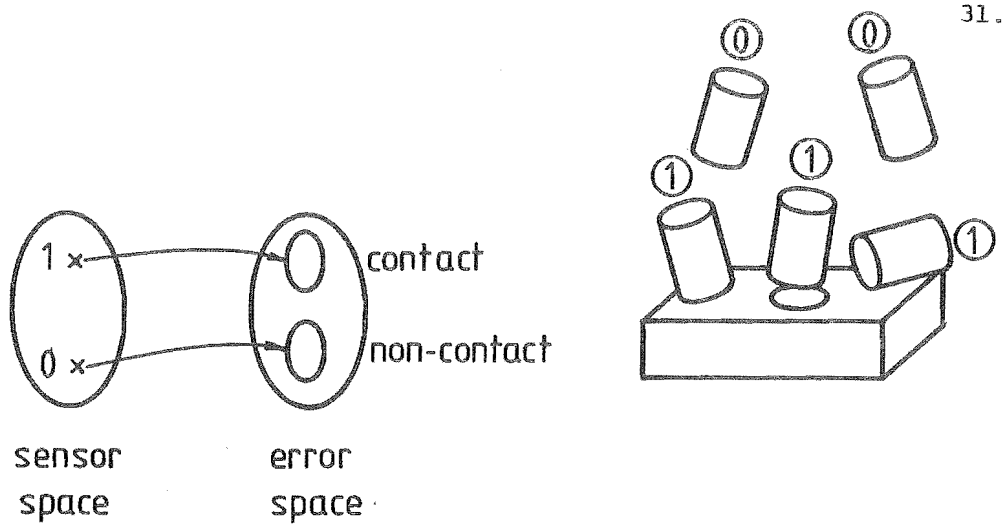
Figure 3.2c illustrates the ambiguity of analogue-vector contact information and provides justification for blind search in the simple peg-hole configuration.

3.2.2 Non-contact sensing.

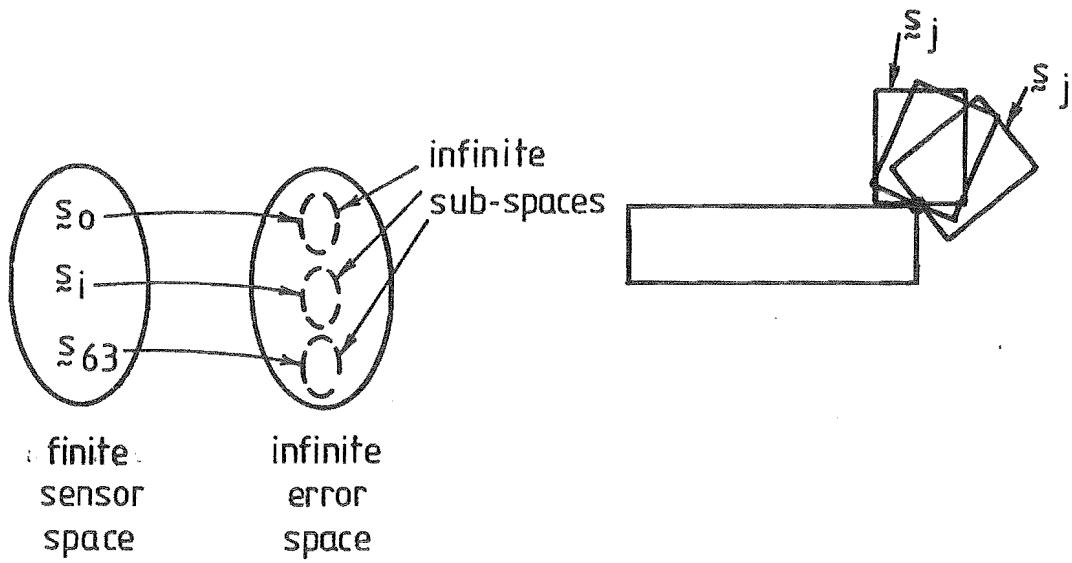
Although various forms of non-contact sensing are possible, this section will only discuss visual sensing, a prevalent form of non-contact sensing found in existing assembly systems. It is assumed that a TV camera is used as the visual organ. A sensor-space vector then consists of light intensity measurements taken at all the points in the image plane of the camera.

First, consider the two-dimensional case where parts are flat and located on a plane surface whose position and orientation with respect to the camera are known.

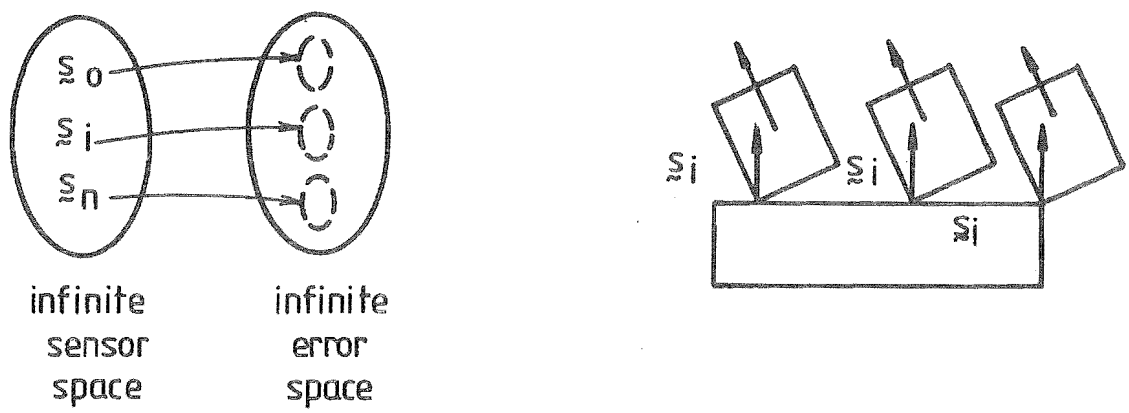
Referring to a pinhole model of the camera ,Fig. 3.3, we observe that each point belonging to an object on the surface projects uniquely onto a point belonging to the image of the object in the camera's image plane, and vice versa, providing of course there is an infinite number of points in the image plane (i.e. the camera is ideal). The co-ordinates of an object point with respect to a set of axes fixed to the camera are then directly proportional to the co-ordinates of the



(a) binary contact sensing



(b) binary-vector contact sensing



(c) analogue-vector contact sensing

FIG 3·2 THE AMBIGUITY OF CONTACT SENSING

corresponding image point. As the coordinates of image points are given by their positions in a sensor-space vector, the coordinates of object points, and hence the location of the object itself, can be readily determined. Clearly, if several objects lie on the plane surface in the field of view of the camera, their relative location can be unambiguously deduced from the sensor-space vector. In other words, visual sensing with an ideal T.V. camera in a two-dimensional situation is a functional relation between sensor and error spaces (Fig.3.4a).

A real camera, however, has its image plane divided into a grid pattern with a finite number of image points where the light intensities from an infinite number of object points are sampled. Thus, there no longer is a one-to-one correspondence between image and object points or a strict functional relation between sensor-space and error-space vectors. Instead, each image point projects into a small portion of the object plane and each sensor-space vector, into a small subset of the error space (Fig. 3.4b). The size of such an error-space sub-set, or object-plane portion, is directly related to the resolution of the camera.

Finally, in three dimensions where objects are not flat or located on a known support, but have thicknesses and may be arbitrarily placed, ambiguity can arise (even with an ideal camera) because any two points on the same light ray passing through the camera's pinhole will share the same image (Fig.3.5a). To avoid this kind of ambiguity, standard methods of obtaining depth information based on the triangulation principle (Fig.3.5b) can be employed; when this is done, spatial visual sensing theoretically becomes a functional relation between sensor and error spaces, although unlike in the two-dimensional case, it is not a binary relation as it relates a pair of sensor-space vectors to every error-space vector. Again, due to the finite resolving power of real cameras, a pair of sensor-space vectors in effect transforms into a hyper-volume of error-space vectors.

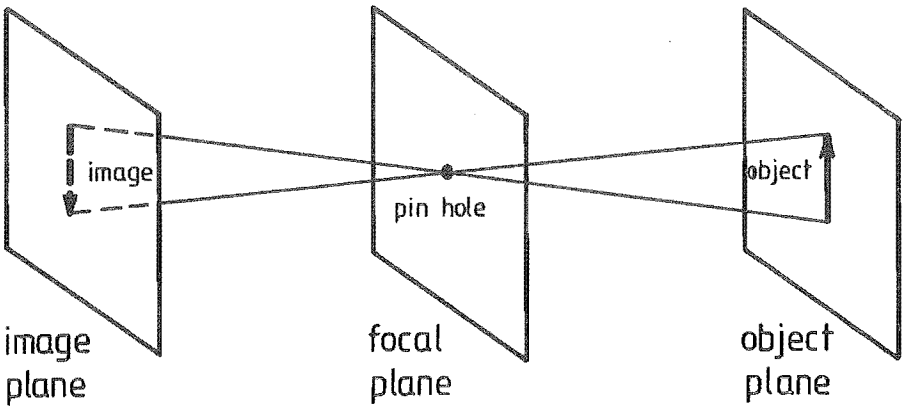


FIG 3·3 PIN HOLE MODEL OF A CAMERA

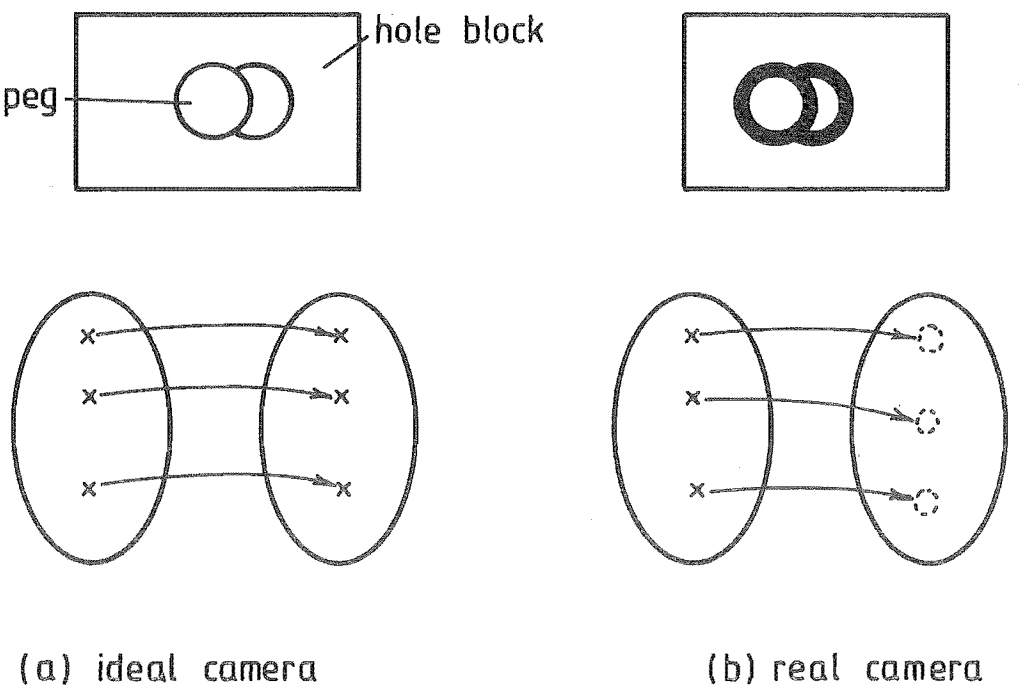
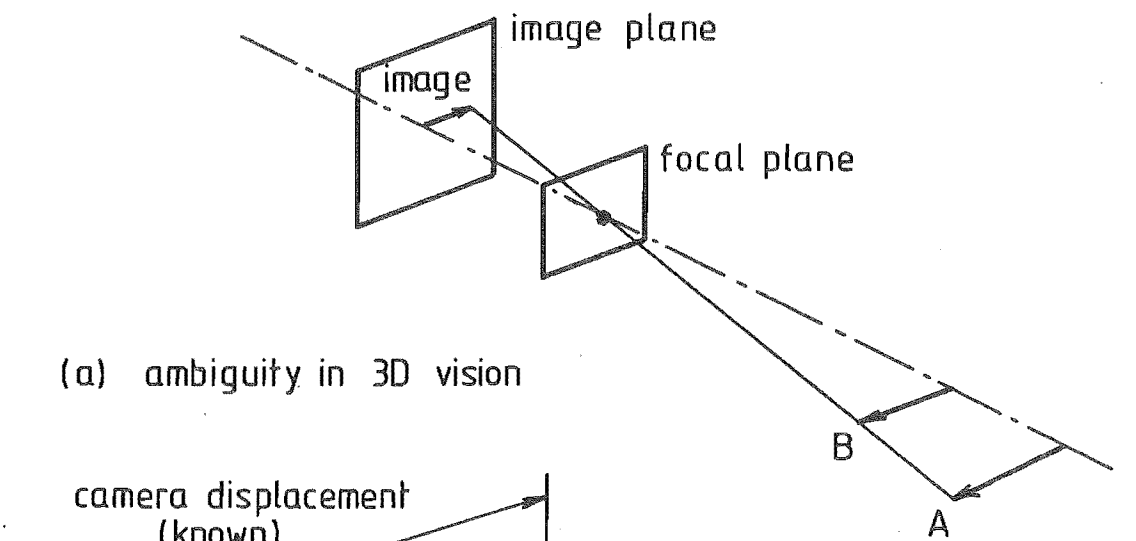
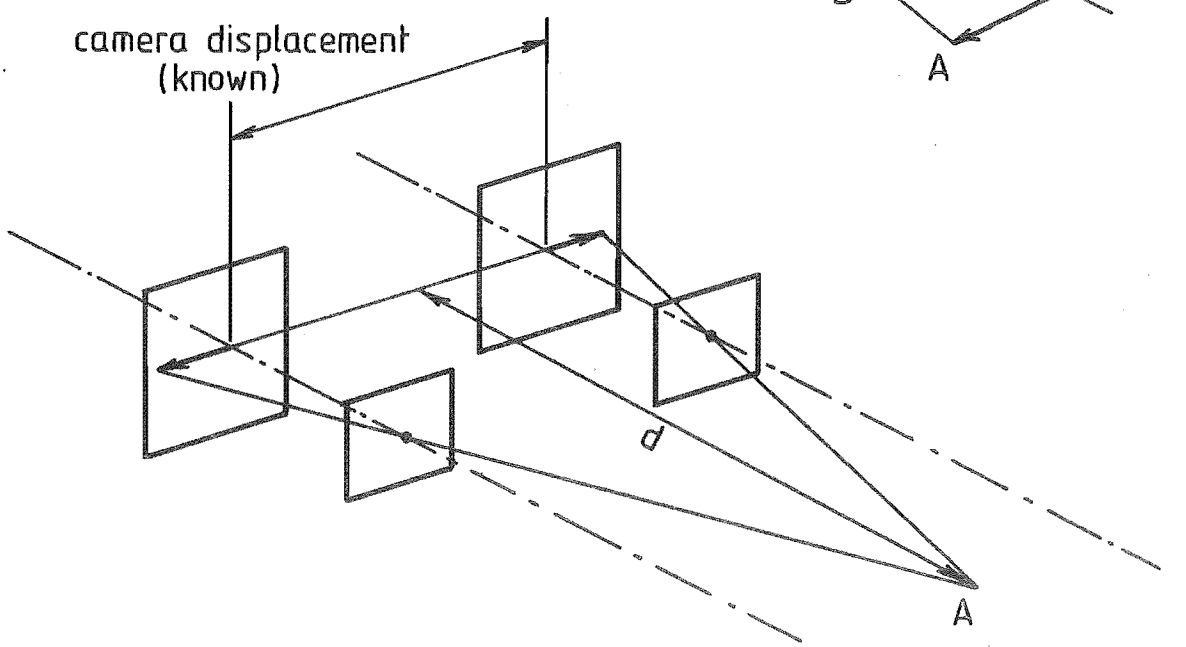


FIG 3·4 2-D VISUAL SENSING



(a) ambiguity in 3D vision



(b) triangulation for finding d

FIG 3·5 3-D VISION

Naturally, in an assembly machine using visual feedback alone, this hyper-volume must be 'smaller' than that allowed by part clearances for direct assembly to occur.

3.2.3 Integrated sensing

Where parts with fine clearances are to be mated, force or touch sensing can be combined with visual sensing to achieve the high resolution required. This combination has been termed 'integrated sensing'.

As mentioned in Chapter 2, integrated sensing is typically carried out in two stages, with visual sensing employed in the first stage to reduce the large initial error space to a 'smaller' hyper-volume, and force or touch sensing in the second stage to implement blind search within this hyper-volume. From Fig. 3.6 it is obvious that integrated sensing cannot be a functional relation between sensor and error spaces.

3.2.4 Discussion

Thus, for various reasons, none of the present sensing strategies is strictly a functional relation and none can give, in a single step and with complete certainty, the state of relative misalignment of two objects during an assembly operation.

The non-functionality of contact sensing is due to its inherent deficiency in vital geometric information. This deficiency, which cannot be remedied by any improvement in contact-sensing hardware, means that the error-space subset associated with each sensor-space vector is large and blind search is mandatory. So as to minimize the error-space subset, and hence the time spent in blind search, it is necessary to restrict the size of the total error space. As previously described, this is achieved in assembly machines equipped with pure contact sensing by structuring the assembly environment so that parts can only deviate from their designed locations by small amounts.

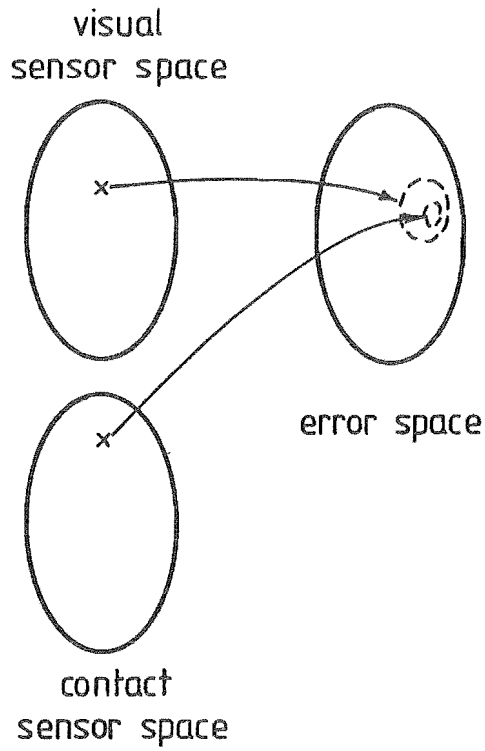


FIG 3·6 INTEGRATED SENSING

The non-functionality of visual sensing on the other hand is mainly caused by the limitations and imperfections of visual hardware. However, since the geometric information content of visual sensing is still fundamentally complete, functionality can be approached as visual hardware is upgraded. Specifically, the error-space subset associated with each sensor-space vector will be contracted, and the definition in a visual image, enhanced, if the ratio of image points and corresponding object points is increased, either by focussing an existing camera onto a narrowed field of view or by using a camera with improved resolution. Although theoretically the image-point-to-object-point ratio can tend to 1, in practice it is limited by the inability of an image processor to cope efficiently with excessive redundant information. (Note that, because only 6 parameters are required to define an error-space vector, a visual-sensor space vector which has several thousand components always contains redundant information. As the resolution of a camera increases, so will the number of components, the amount of redundancy, and hence the time required for sorting out useful geometric information.)

Finally, integrated sensing is a non-functional relation because it is a nested sequence of two non-functional relations. Integrated sensing machines can also be regarded as contact-sensing machines in which the size of the total error space is controlled by visual sensing rather than by environmental structuring.

3.3 BILATERAL CONTACT SENSING

Basically, contact sensors are inexpensive and contact sensing, easy to implement. However, it has been found that unilateral contact sensing must usually be complemented by either visual sensing or environmental structuring to limit the total error space and facilitate blind search. Some of the penalties then incurred are the complexity of artificial vision or the inflexibility associated with environmental structuring.

On the other hand, if contact sensing can be complemented by contact sensing, its inherent low cost and simplicity will be retained and the above penalties averted. This is the case with bilateral (or stereo) contact sensing where two arrays of contact sensors are employed and the information related to contact geometry during the assembly of two objects is extracted simultaneously from their respective arrays.

In this section we shall outline the general principle of bilateral sensing and illustrate it with examples of a family of bilateral contact sensing techniques.

3.3.1 Principle

Let A_1 and A_2 be two sensor arrays fixed respectively to two rigid bodies B_1 and B_2 whose relative location $\underset{\sim}{e}_B$ is to be determined. Clearly, $\underset{\sim}{e}_B$ can be computed from the relative location $\underset{\sim}{e}_A$ of A_1 and A_2 provided $\underset{\sim}{e}_A$ is known.

Bilateral sensing is an indirect technique for obtaining $\underset{\sim}{e}_A$ based on comparing the observations of A_1 and A_2 when these arrays are stimulated by identical world events (e.g. light rays from a point source or contact reactions on B_1 and B_2 at their mutual interface.)

To appreciate how comparing the observations of A_1 and A_2 can yield $\underset{\sim}{e}_A$, consider the special case where the observations of A_1 coincide with those of A_2 for all world events: it is obvious that A_1 and A_2 must then coincide, or $\underset{\sim}{e}_A = \underset{\approx}{0}$.

More generally, if $\underset{\sim}{s}_1$ and $\underset{\sim}{s}_2$ are the observations of A_1 and A_2 corresponding to the same world event W, comparing $\underset{\sim}{s}_1$ and $\underset{\sim}{s}_2$ becomes equivalent to intersecting their respective error-space subsets E_1 and E_2 (Fig. 3.7a). Since by definition E_1 and E_2 each contains $\underset{\sim}{e}_A$, the intersection of E_1 and E_2 denoted by $E_1 \cap E_2$, must also contain $\underset{\sim}{e}_A$.

The problem of inferring $\underset{\sim}{e}_A$ from $E_1 \cap E_2$ would be trivial if $E_1 \cap E_2$ were

an ideal intersection of E_1 and E_2, that is, if this intersection contained only one element - the vector $\underset{\sim}{e}_A$. However, even when $E_1 \cap E_2$ contains other elements in addition to $\underset{\sim}{e}_A$, we can still intuitively visualize the possibility of pinpointing $\underset{\sim}{e}_A$ if the intersection of error-space subsets is repeated ad infinitum (Fig 3.7b). In practice, the number of intersections then required is finite and dependent on the exact nature of each intersection.

3.3.2 A family of bilateral contact sensing techniques

How multiple intersections of error-space subsets can yield $\underset{\sim}{e}_A$ will now be demonstrated mathematically for cases where both A_1 and A_2 are arrays of contact sensors.

In the following discussion, it is assumed that A_1 is mounted on a robot hand and A_2 on a work table. (Thus body B_1 can be the robot hand or a part thereof and similarly, body B_2 can be the work table or a part thereof.) Let the location of A_1 relative to some fixed reference be defined by the triad of coordinate axes $(\underset{\sim}{X}_1, \underset{\sim}{Y}_1, \underset{\sim}{Z}_1)$ and the location of A_2, by $(\underset{\sim}{X}_2, \underset{\sim}{Y}_2, \underset{\sim}{Z}_2)$ as shown in Fig. 3.8. For simplicity we also take $(\underset{\sim}{X}_1, \underset{\sim}{Y}_1, \underset{\sim}{Z}_1)$ as representing the location of B_1, and $(\underset{\sim}{X}_2, \underset{\sim}{Y}_2, \underset{\sim}{Z}_2)$, the location of B_2.

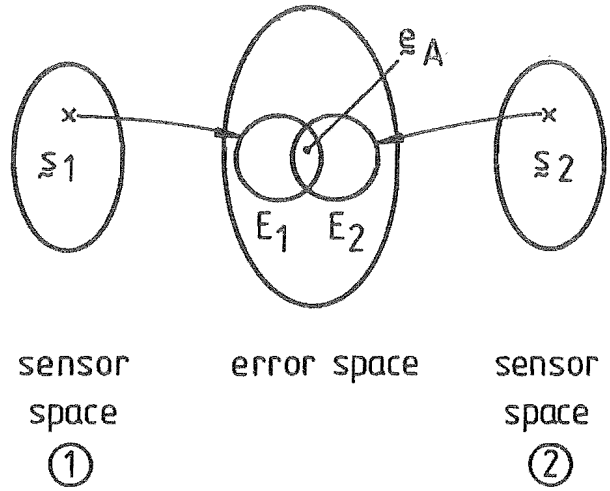
As mentioned previously, the location of A_2 relative to A_1 is specified by $\underset{\sim}{e}_A$ where

$$\underset{\sim}{e}_A = \begin{bmatrix} \underset{\sim}{t} \\ \hline \underset{\sim}{r} \end{bmatrix}$$

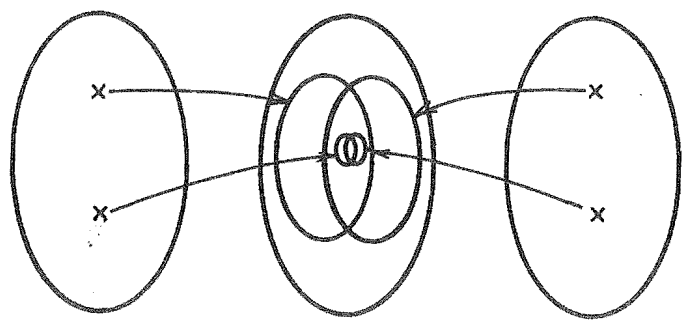
$\underset{\sim}{t}$ being a translation vector which defines the position of A_2 relative to A_1, and $\underset{\sim}{r}$, a rotation 'vector' which defines the orientation of A_2 relative to A_1.

Generally $\underset{\sim}{t} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ where x, y and z are the coordinates of the origin of $(\underset{\sim}{X}_2, \underset{\sim}{Y}_2, \underset{\sim}{Z}_2)$ in $(\underset{\sim}{X}_1, \underset{\sim}{Y}_1, \underset{\sim}{Z}_1)$.

However, several alternative forms for $\underset{\sim}{r}$ exist, including



(a) the intersection of error-space subsets



(b) the double intersection of error-space subsets

FIG 3·7 THE PRINCIPAL OF BILATERAL SENSING

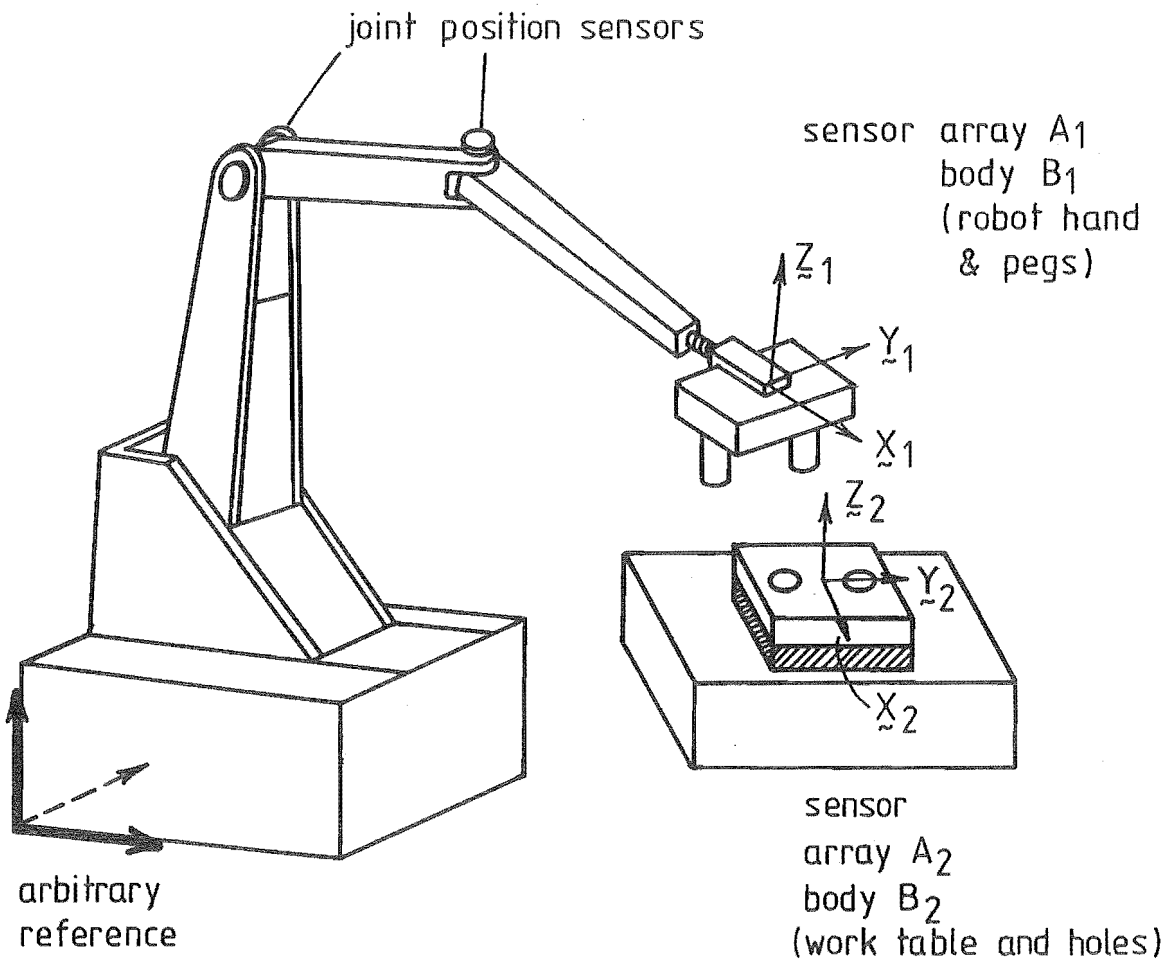


FIG 3·8 A TYPICAL ASSEMBLY SYSTEM

$\underset{\sim}{r}_E = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$ where ϕ, θ and ψ are the Euler angles of $(\underset{\sim}{X}_2, \underset{\sim}{Y}_2, \underset{\sim}{Z}_2)$ relative to $(\underset{\sim}{X}_1, \underset{\sim}{Y}_1, \underset{\sim}{Z}_1)$,

$\underset{\sim}{r}_C = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}$ where θ_x, θ_y and θ_z are the angles of the elementary rotations R_x, R_y and R_z about the Cartesian axes $\underset{\sim}{X}_1$, $\underset{\sim}{Y}_1$ and $\underset{\sim}{Z}_1$, which, when combined in this order, transform the orientation of A_1 into that of A_2.

and $\underset{\sim}{r}_R = \begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix}$ where λ, μ and ν are the Rodrigues parameters corresponding to the single rotation which transforms the orientation of A_1 into that of A_2.

For convenience, we shall not deal with any of these particular forms of $\underset{\sim}{r}$ in this Chapter. Instead, we shall be concerned only with finding the general rotation matrix [R] which also completely specifies the orientation of A_2 with respect to A_1. [R] is a 3x3 orthogonal matrix defined as

$$[R] \equiv [\hat{\underset{\sim}{n}}_i \quad \hat{\underset{\sim}{n}}_j \quad \hat{\underset{\sim}{n}}_k] \qquad \text{Eq. 3.0}$$

where the column vectors $\hat{\underset{\sim}{n}}_i$, $\hat{\underset{\sim}{n}}_j$ and $\hat{\underset{\sim}{n}}_k$ are precisely $\underset{\sim}{X}_2$, $\underset{\sim}{Y}_2$ and $\underset{\sim}{Z}_2$ expressed in the $(\underset{\sim}{X}_1, \underset{\sim}{Y}_1, \underset{\sim}{Z}_1)$ coordinate system.

$\hat{\underset{\sim}{n}}_i$, $\hat{\underset{\sim}{n}}_j$ and $\hat{\underset{\sim}{n}}_k$ satisfy the following identities

$$\hat{\underset{\sim}{n}}_i^2 = 1 \qquad \text{Eq. 3.1}$$

$$\hat{\underset{\sim}{n}}_j^2 = 1 \qquad \text{Eq. 3.2}$$

$$\hat{\underset{\sim}{n}}_i \cdot \hat{\underset{\sim}{n}}_j = 0 \qquad \text{Eq. 3.3}$$

$$\hat{\underset{\sim}{n}}_k - \hat{\underset{\sim}{n}}_i \times \hat{\underset{\sim}{n}}_j = 0 \qquad \text{Eqs 3.4 to 3.6}$$

which are the orthogonality conditions for [R].

Note that once [R] is known, $\underset{\sim}{r}_E$, $\underset{\sim}{r}_C$ and $\underset{\sim}{r}_R$ can be easily derived since

$$[R] = \begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z - \cos\theta_x \sin\theta_z & \cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z \\ \cos\theta_y \sin\theta_z & \sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix}$$

$$= \frac{1}{\Delta} \begin{bmatrix} 1 + \frac{1}{4}(\lambda^2 - \mu^2 - \nu^2) & -\nu + \frac{1}{2}\lambda\mu & \mu + \frac{1}{2}\lambda\nu \\ \nu + \frac{1}{2}\lambda\mu & 1 + \frac{1}{4}(-\lambda^2 + \mu^2 - \nu^2) & -\lambda + \frac{1}{2}\mu\nu \\ -\mu + \frac{1}{2}\lambda\nu & \lambda + \frac{1}{2}\mu\nu & 1 + \frac{1}{4}(-\lambda^2 - \mu^2 + \nu^2) \end{bmatrix}$$

where $\Delta = 1 + \frac{1}{4}(\lambda^2 + \mu^2 + \nu^2)$

We shall now describe four groups of techniques for deriving $\underset{\sim}{t}$ and [R]. These groups have been arranged according to whether

1 - A_1 and A_2 are both force sensor arrays

2 - A_1 is made up of touch sensors, and A_2 of force sensors

3 - A_1 is made up of force sensors, and A_2 of touch sensors

or 4 - A_1 and A_2 are both touch sensor arrays.

3.3.2.1 Force-force sensing

It is assumed that both the robot hand (body B_1) and the work table (body B_2) are equipped with force sensors (arrays A_1 and A_2). First, suppose B_1 and B_2 are in static contact and the forces and moments they mutually exert are picked up by A_1 and A_2 as $(\underset{\sim}{f}_1, \underset{\sim}{m}_1)$ and $(\underset{\sim}{f}_2, \underset{\sim}{m}_2)$. For equilibrium (see also Fig. 3.9a)

$$\underset{\sim}{f}_1 = - [R] \underset{\sim}{f}_2 \qquad \text{Eqs. 3.7 to 3.9}$$

$$\underset{\sim}{m}_1 + [R] \underset{\sim}{m}_2 = \underset{\sim}{t} \times \underset{\sim}{f}_1 \qquad \text{Eqs. 3.10 to 3.12}$$

Suppose now that the contact forces and moments mutually exerted by B_1 and B_2 are changed to $(\underset{\sim}{F}_1, \underset{\sim}{M}_1)$ and $(\underset{\sim}{F}_2, \underset{\sim}{M}_2)$. Again $(\underset{\sim}{F}_1, \underset{\sim}{M}_1)$ and $(\underset{\sim}{F}_2, \underset{\sim}{M}_2)$ will satisfy the following equilibrium relations

$$\underset{\sim}{F}_1 = - [R] \underset{\sim}{F}_2 \qquad \text{Eqs 3.13 to 3.15}$$

$$\underset{\sim}{M}_1 + [R] \underset{\sim}{M}_2 = \underset{\sim}{t} \times \underset{\sim}{F}_1 \qquad \text{Eqs 3.16 to 3.18}$$

The set of Equations 3.0 to 3.18 is solvable for [R] and $\underset{\sim}{t}$ if $\underset{\sim}{f}_1$ and $\underset{\sim}{F}_1$ are linearly independent. An exact, closed-form solution based on vector algebra exists and will be described in Chapter 4. Another method of finding [R] and $\underset{\sim}{t}$ involves numerical procedures. For example, Brown's algorithm for solving systems of non-linear equations [77] (which will be explained in Chapter 5) can be applied to the set of Equations 3.0 to 3.8 and 3.13, to yield [R]. $\underset{\sim}{t}$ can then be obtained from any triplet of linearly-independent equations selected from the set of Equations 3.10 to 3.12 and 3.16 to 3.18.

In Equations 3.13 to 3.18, for simplicity we have assumed that the relative location between B_1 and B_2 remains constant although the contact forces and moments change from $(\underset{\sim}{f}_1, \underset{\sim}{m}_1)$ and $(\underset{\sim}{f}_2, \underset{\sim}{m}_2)$ to $(\underset{\sim}{F}_1, \underset{\sim}{M}_1)$ and $(\underset{\sim}{F}_2, \underset{\sim}{M}_2)$. However, in reality so that $\underset{\sim}{F}_1$ and $\underset{\sim}{f}_1$ can be linearly independent, the relative location between B_1 and B_2 must change. We shall now demonstrate

that as long as this change is known, the new system of equations obtained will still be of the same form as above.

Let the (known) displacement of B_1 be represented by $[R]_D$ and $\underset{\sim}{t}_D$ and the new location of B_2 relative to B_1, by $[R]_{NEW}$ and $\underset{\sim}{t}_{NEW}$ (Fig.3.9b). If the forces and moments measured by A_1 and A_2 are again ($\underset{\sim}{F}_1$, $\underset{\sim}{M}_1$) and ($\underset{\sim}{F}_2$, $\underset{\sim}{M}_2$), then

$$\underset{\sim}{F}_1 = -[R]_{NEW}\ \underset{\sim}{F}_2 \qquad \text{Eqs 3.19 to 3.21}$$

$$\underset{\sim}{M}_1 + [R]_{NEW}\ \underset{\sim}{M}_2 = \underset{\sim}{t}_{NEW} \times \underset{\sim}{F}_1 \qquad \text{Eqs 3.22 to 3.24}$$

From McCallion and Pham [78]

$$[R]_{NEW} = [R]_D^T\ [R]$$

$$\underset{\sim}{t}_{NEW} = [R]_D^T\ (\underset{\sim}{t} - \underset{\sim}{t}_D)$$

where $[R]_D^T$ is the transpose as well as the inverse of $[R]_D$.

Substituting these expressions for $[R]_{NEW}$ and $\underset{\sim}{t}_{NEW}$ in Eqs 3.19 to 3.21 gives

$$\underset{\sim}{F}_1 = -[R]_D^T\ [R]\ \underset{\sim}{F}_2 \qquad \text{Eqs 3.25 to 3.27}$$

$$\underset{\sim}{M}_1 + [R]_D^T\ [R]\ \underset{\sim}{M}_2 = \left([R]_D^T\ (\underset{\sim}{t} - \underset{\sim}{t}_D)\right) \times \underset{\sim}{F}_1 \qquad \text{Eqs 3.28 to 3.30}$$

Multiplying both sides of Eqs 3.25 to 3.27 by $[R]_D$ gives

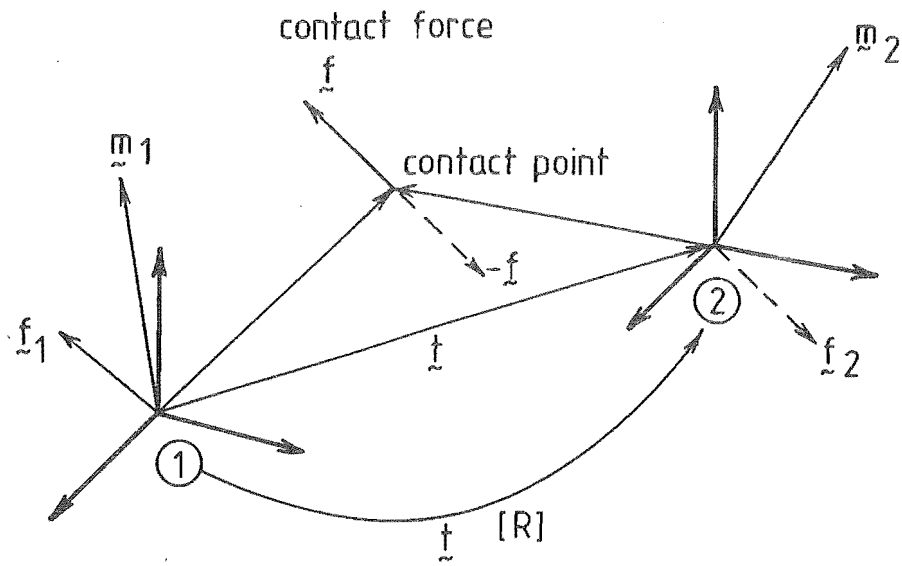
$$[R]_D\ \underset{\sim}{F}_1 = -[R]\ \underset{\sim}{F}_2 \qquad \text{Eqs 3.31 to 3.33}$$

Let $\underset{\sim}{F}_1^* = [R]_D\ \underset{\sim}{F}_1$. Since $[R]_D$ and $\underset{\sim}{F}_1$ are known, $\underset{\sim}{F}_1^*$ is also known. Eqs 3.31 to 3.33 become

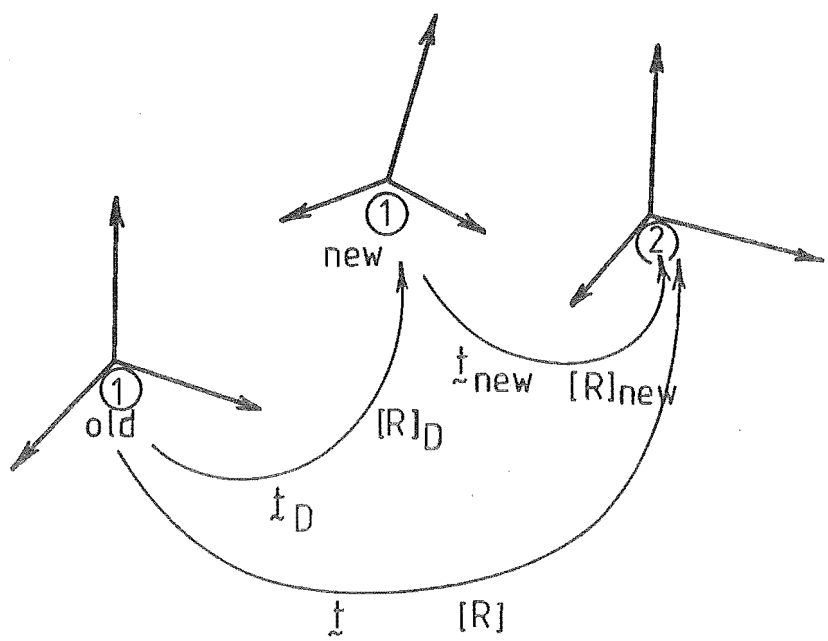
$$\underset{\sim}{F}_1^* = -[R]\ \underset{\sim}{F}_2 \qquad \text{Eqs 3.34 to 3.36}$$

which are identical in form to Eqs 3.13 to 3.15.

Now, substituting the expression for $\underset{\sim}{F}_1$ (Eqs 3.25 to 3.27) in Eqs 3.28 to 3.30 gives



(a) equilibrium of forces



(b) coordinate transformation

FIG 3·9 FORCE-FORCE SENSING

$$\underset{\sim}{M}_1 + [R]_D^T [R] \underset{\sim}{M}_2 = - \left([R]_D^T (\underset{\sim}{t} - \underset{\sim}{t}_D) \right) \times [R]_D^T [R] \underset{\sim}{F}_2 \qquad \text{Eqs 3.37 to 3.39}$$

As $[R]_D^T$ is orthogonal, $[R]_D^T \underset{\sim}{x} \times [R]_D^T \underset{\sim}{y} = [R]_D^T (\underset{\sim}{x} \times \underset{\sim}{y})$ for any $\underset{\sim}{x}$ and $\underset{\sim}{y}$. Thus Eqs 3.37 to 3.39 become

$$\underset{\sim}{M}_1 + [R]_D^T [R] \underset{\sim}{M}_2 = - [R]_D^T [(\underset{\sim}{t} - \underset{\sim}{t}_D) \times [R] \underset{\sim}{F}_2] \qquad \text{Eqs 3.40 to 3.42}$$

Replacing $[R] \underset{\sim}{F}_2$ by $-\underset{\sim}{F}_1^*$ in Eqs 3.40 to 3.42, multiplying both sides of these equations by $[R]_D$, and rearranging the result, give

$$[R]_D \underset{\sim}{M}_1 + \underset{\sim}{t}_D \times \underset{\sim}{F}_1^* + [R] \underset{\sim}{M}_2 = \underset{\sim}{t} \times \underset{\sim}{F}_1^* \qquad \text{Eqs 3.43 to 3.45}$$

Let $\underset{\sim}{M}_1^* = [R]_D \underset{\sim}{M}_1 + \underset{\sim}{t}_D \times \underset{\sim}{F}_1^*$

Since $[R]_D$, $\underset{\sim}{M}_1$, $\underset{\sim}{t}_D$, and $\underset{\sim}{F}_1^*$ are known, $\underset{\sim}{M}_1^*$ is also known. Eqs 3.43 to 3.45 become

$$\underset{\sim}{M}_1^* + [R] \underset{\sim}{M}_2 = \underset{\sim}{t} \times \underset{\sim}{F}_1^* \qquad \text{Eqs 3.46 to 3.48}$$

which are identical in form to Eqs 3.16 to 3.18.

Thus the methods for computing [R] and $\underset{\sim}{t}$ from the set of equations 3.0 to 3.18 can equally be applied for computing [R] and $\underset{\sim}{t}$ from the set of equations 3.0 to 3.6, 3.34 to 3.36, and 3.46 to 3.48.

3.3.2.2 Touch-force sensing

Here, it is assumed that the robot hand (body B_1) is fitted with a 'wand' (array A_1) which can detect when touching occurs at its sensitized tip. The wand is rigid and the location of its tip, or in other words the position of the contact point, with respect to the robot hand coordinate system is known. The work table (body B_2) is again assumed to have an array of force sensors (array A_2) capable of measuring forces and moments in three dimensions.

We shall now indicate how [R] and $\underset{\sim}{t}$ can be found if the robot establishes

3 non-collinear contact points with the work table. There are two cases to consider, depending on whether the positions of the contact points are individually or collectively determined in the work table coordinate system.

(It is recalled that the coordinate systems of B_1 and B_2 have been taken as coinciding with those of A_1 and A_2 respectively.)

(i) Contact points found individually

Let $\underset{\sim}{U}_1$ be the known position vector, in the robot hand coordinate system, of the tip of the wand when it touches the work table at a point U. The position vector $\underset{\sim}{U}_2$ of the contact point U in the work table system may be found, for example, by intersecting the surface S of the work table with the line of action of the contact force exerted at U by the wand tip on the work table (this line of action is completely determinable from the force and moment information picked up by A_2).

Similarly, let $\underset{\sim}{V}_1$ and $\underset{\sim}{W}_1$ be the known position vectors, in the robot hand coordinate system, of the tip of the wand when it touches the work table at V and W. Again, the position vectors $\underset{\sim}{V}_2$ and $\underset{\sim}{W}_2$ of V and W in the work table system may be found by intersecting S with the lines of action of the contact forces through V and W respectively (Fig. 3.10).

The following rules for coordinate transformation apply

$$\underset{\sim}{U}_1 = [R]\ \underset{\sim}{U}_2 + \underset{\sim}{t} \qquad \text{Eqs 3.49 to 3.51}$$

$$\underset{\sim}{V}_1 = [R]\ \underset{\sim}{V}_2 + \underset{\sim}{t} \qquad \text{Eqs 3.52 to 3.54}$$

$$\underset{\sim}{W}_1 = [R]\ \underset{\sim}{W}_2 + \underset{\sim}{t} \qquad \text{Eqs 3.55 to 3.57}$$

Therefore

$$\underset{\sim}{U}_1 - \underset{\sim}{V}_1 = [R]\ (\underset{\sim}{U}_2 - \underset{\sim}{V}_2) \qquad \text{Eqs 3.58 to 3.60}$$

$$\underset{\sim}{U}_1 - \underset{\sim}{W}_1 = [R]\ (\underset{\sim}{U}_2 - \underset{\sim}{W}_2) \qquad \text{Eqs 3.61 to 3.63}$$

which are identical in form to Eqs 3.7 to 3.9 and Eqs 3.13 to 3.15. If U, V and W are non-collinear, $(\underset{\sim}{U}_1 - \underset{\sim}{V}_1)$ and $(\underset{\sim}{U}_1 - \underset{\sim}{W}_1)$ are linearly

independent and [R] can be found as indicated in Section 3.3.2.1. Then either Eqs 3.49 to 3.51, Eqs 3.52 to 3.54, or Eqs 3.55 to 3.57 can be employed to give $\underset{\sim}{t}$ directly.

(ii) Contact points found collectively

When an accurate model of the surface S is not available, $\underset{\sim}{U}_2$, $\underset{\sim}{V}_2$ and $\underset{\sim}{W}_2$ may be found collectively, as we shall see below.

Again, we assume that, using the wand, the robot makes three successive contacts with the work table at 3 non-collinear points, U, V and W. Let ℓ_u, ℓ_v, and ℓ_w be the lines of action of the contact forces through these points. In the work table coordinate system, ℓ_u, ℓ_v and ℓ_w can be described by $\underset{\sim}{L}_{u2}$, $\underset{\sim}{L}_{v2}$ and $\underset{\sim}{L}_{w2}$, such that

$$\underset{\sim}{L}_{u2} = \alpha \underset{\sim}{F}_{u2} + \frac{\underset{\sim}{F}_{u2} \times \underset{\sim}{M}_{u2}}{\underset{\sim}{F}_{u2}^{\,2}} \qquad \text{Eqs 3.64 to 3.66}$$

$$\underset{\sim}{L}_{v2} = \beta \underset{\sim}{F}_{v2} + \frac{\underset{\sim}{F}_{v2} \times \underset{\sim}{M}_{v2}}{\underset{\sim}{F}_{v2}^{\,2}} \qquad \text{Eqs 3.67 to 3.69}$$

$$\underset{\sim}{L}_{w2} = \gamma \underset{\sim}{F}_{w2} + \frac{\underset{\sim}{F}_{w2} \times \underset{\sim}{M}_{w2}}{\underset{\sim}{F}_{w2}^{\,2}} \qquad \text{Eqs 3.70 to 3.72}$$

where $\underset{\sim}{L}_{u2}$, $\underset{\sim}{L}_{v2}$ and $\underset{\sim}{L}_{w2}$ are the position vectors, in the work table system of arbitrary points on ℓ_u, ℓ_v, and ℓ_w; α, β, and γ are arbitrary scalars; $(\underset{\sim}{F}_{u2}, \underset{\sim}{M}_{u2})$, $(\underset{\sim}{F}_{v2}, \underset{\sim}{M}_{v2})$ and $(\underset{\sim}{F}_{w2}, \underset{\sim}{M}_{w2})$ are the forces and moments felt by A_2, due to the contact forces through U, V and W.

Since U, V and W are located on ℓ_u, ℓ_v and ℓ_w, $\underset{\sim}{U}_2$, $\underset{\sim}{V}_2$ and $\underset{\sim}{W}_2$ must satisfy Eqs 3.64 to 3.72, i.e.

$$\underset{\sim}{U}_2 = \alpha^* \underset{\sim}{F}_{u2} + \frac{\underset{\sim}{F}_{u2} \times \underset{\sim}{M}_{u2}}{\underset{\sim}{F}_{u2}^{\,2}} \qquad \text{Eqs 3.73 to 3.75}$$

$$\underset{\sim}{V}_2 = \beta^* \underset{\sim}{F}_{v2} + \frac{\underset{\sim}{F}_{v2} \times \underset{\sim}{M}_{v2}}{\underset{\sim}{F}_{v2}^{\,2}} \qquad \text{Eqs 3.76 to 3.78}$$

$$\underset{\sim}{W}_2 = \gamma^* \underset{\sim}{F}_{w2} + \frac{\underset{\sim}{F}_{w2} \times \underset{\sim}{M}_{w2}}{\underset{\sim}{F}_{w2}^2} \qquad \text{Eqs 3.79 to 3.81}$$

α^*, β^* and γ^* can be obtained simultaneously from the set of non-linear equations which express the invariance of scalar distances in coordinate transformations, viz:

$$(\underset{\sim}{U}_2 - \underset{\sim}{V}_2)^2 = (\underset{\sim}{U}_1 - \underset{\sim}{V}_1)^2 \qquad \text{Eq. 3.82}$$

$$(\underset{\sim}{U}_2 - \underset{\sim}{W}_2)^2 = (\underset{\sim}{U}_1 - \underset{\sim}{W}_1)^2 \qquad \text{Eq. 3.83}$$

$$(\underset{\sim}{V}_2 - \underset{\sim}{W}_2)^2 = (\underset{\sim}{V}_1 - \underset{\sim}{W}_1)^2 \qquad \text{Eq. 3.84}$$

where $\underset{\sim}{U}_1$, $\underset{\sim}{V}_1$, and $\underset{\sim}{W}_1$ are the position vectors of U, V, and W in the robot hand coordinate system.

Again, Brown's algorithm may be applied to solve Eqs 3.82 to 3.84 for α^*, β^* and γ^*. Once α^*, β^* and γ^* are found, $\underset{\sim}{U}_2$, $\underset{\sim}{V}_2$ and $\underset{\sim}{W}_2$ are known (Eqs 3.73 to 3.81) and the method described in (i) may be used to determine [R] and $\underset{\sim}{t}$.

3.3.2.3 Force-touch sensing

In this case the robot hand is still equipped with a wand. However, instead of detecting binary contact at the tip of the wand, the robot will determine the direction of any contact force applied there, using force sensors located in its hand. A small ball bearing fitted to the tip of the wand ensures frictionless contacts with the work table.

It is assumed that the work table has three smooth, mutually orthogonal datum planes P_i, P_j and P_k, on which these contacts will be made. Touch sensors are fixed to P_i, P_j and P_k to inform the robot on the identity of the plane it is touching.

For convenience, the normals to P_i, P_j and P_k are taken as the axes of the work table coordinate system, and the point of intersection of P_i,

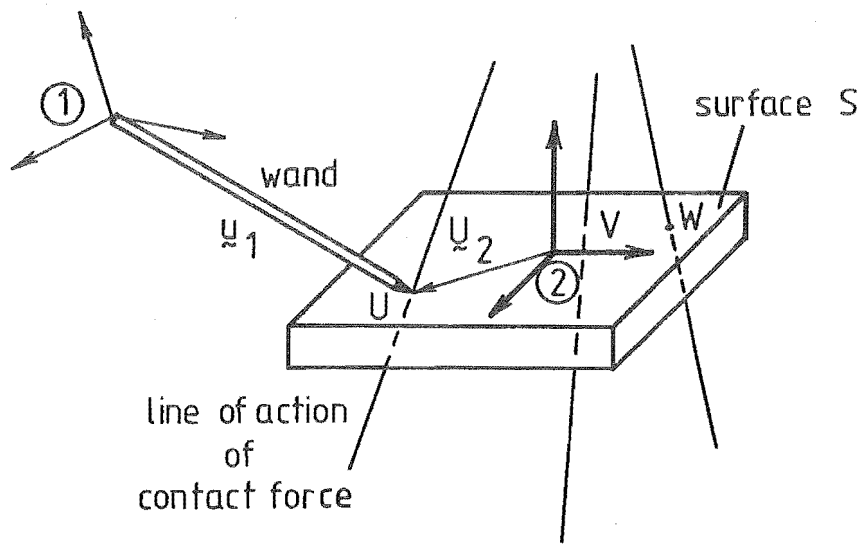


FIG 3·10 TOUCH-FORCE SENSING

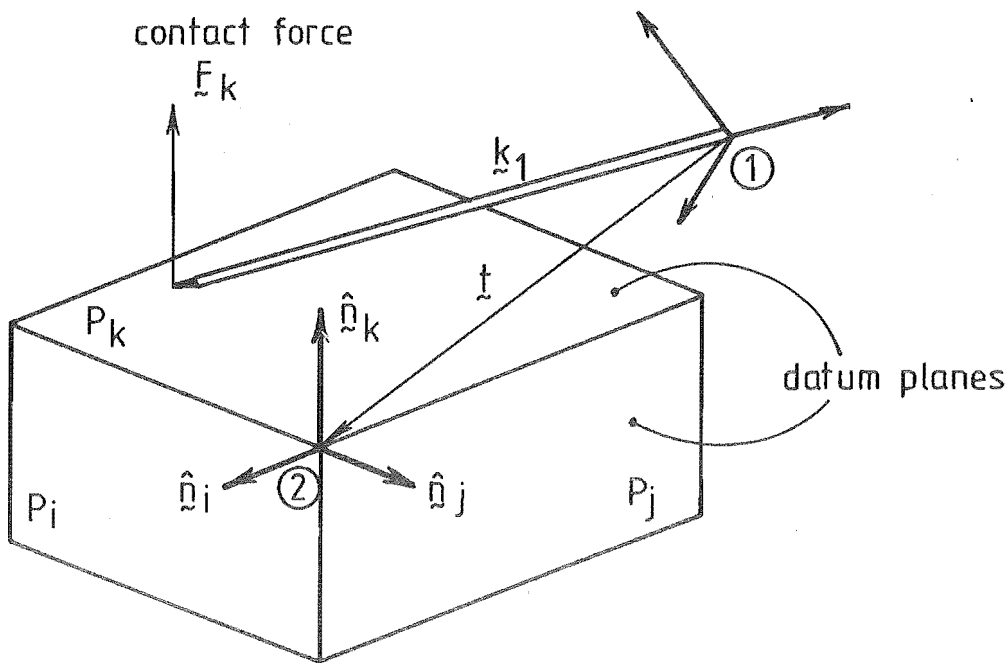


FIG 3·11 FORCE-TOUCH SENSING

P_j and P_k, as the origin of this sytem (Fig. 3.11).

Relative to the robot-hand coordinate system, the normals are

$$\hat{n}_{\sim i} = \frac{F_{\sim il}}{F_{il}}$$

$$\hat{n}_{\sim j} = \frac{F_{\sim jl}}{F_{jl}}$$

$$\hat{n}_{\sim k} = \frac{F_{\sim kl}}{F_{kl}}$$

where $F_{\sim il}$, $F_{\sim jl}$ and $F_{\sim kl}$ are the contact forces on the tip of the wand when it touches P_i, P_j and P_k respectively.

Hence, from Eq. 3.0,

$$[R] = \begin{bmatrix} \frac{F_{\sim il}}{F_{il}} & \frac{F_{\sim jl}}{F_{jl}} & \frac{F_{\sim kl}}{F_{kl}} \end{bmatrix}$$

Let $i_{\sim l}$, $j_{\sim l}$, and $k_{\sim l}$ be the position vectors of the tip of the wand when it touches P_i, P_j and P_k respectively. The position vector $t_{\sim}$ of the point of intersection of P_i, P_j and P_k must satisfy the following relations

$$\hat{n}_{\sim i} \cdot (t_{\sim} - i_{\sim l}) = 0 \qquad \text{Eq. 3.85}$$

$$\hat{n}_{\sim j} \cdot (t_{\sim} - j_{\sim l}) = 0 \qquad \text{Eq. 3.86}$$

$$\hat{n}_{\sim k} \cdot (t_{\sim} - k_{\sim l}) = 0 \qquad \text{Eq. 3.87}$$

since $(t_{\sim} - i_{\sim l})$, $(t_{\sim} - j_{\sim l})$ and $(t_{\sim} - k_{\sim l})$ are parallel to P_i, P_j and P_k respectively (see Fig. 3.11).

Equations 3.85 to 3.87 can be rewritten as

$$\begin{bmatrix} \hat{n}_{\sim i} & \hat{n}_{\sim j} & \hat{n}_{\sim k} \end{bmatrix}^T t_{\sim} = C_{\sim} \qquad \text{Eqs 3.88 to 3.90}$$

where

$$C_{\sim} = \begin{pmatrix} \hat{n}_{\sim i} \cdot i_{\sim l} \\ \hat{n}_{\sim j} \cdot j_{\sim l} \\ \hat{n}_{\sim k} \cdot k_{\sim l} \end{pmatrix}$$

$$\text{or} \quad [R]^T \underset{\sim}{t} = \underset{\sim}{C}$$

$$\therefore \quad \underset{\sim}{t} = [R] \underset{\sim}{C}$$

3.3.2.4 Touch-touch sensing

The robot hand again has a wand with a tip sensitive to touching as in Section 3.3.2.2. The work table retains the datum-plane and touch-sensor arrangement of Section 3.3.2.3. No force sensors are involved.

We shall describe three strategies for deriving [R] and $\underset{\sim}{t}$, based solely on knowledge of the position of the wand tip when it contacts the datum planes. The first strategy involves the robot in contacting each of the three planes at two points. In the second strategy, three contact points are established with one plane, two with a second plane, and one with the third plane. Both these strategies utilise the minimum possible number of contact points but will not work if certain geometric conditions hold between the relative locations of these points. To avoid complications a redundant contact point is introduced in the third strategy where one contact point is made with one plane, and three with each of the other two planes.

(i) First strategy : two contact points on each of P_i, P_j and P_k.

Let $\underset{\sim}{i}_1$, $\underset{\sim}{i}_1'$, $\underset{\sim}{j}_1$, $\underset{\sim}{j}_1'$, $\underset{\sim}{k}_1$ and $\underset{\sim}{k}_1'$ be the position vectors, relative to the robot hand coordinate system, of the tip of the wand when it contacts the datum planes P_i, P_j and P_k respectively.

Since $(\underset{\sim}{i}_1 - \underset{\sim}{i}_1')$, $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1')$, $(\underset{\sim}{k}_1 - \underset{\sim}{k}_1')$ are parallel to P_i, P_j and P_k respectively (see Fig. 3.12a)

$$\hat{\underset{\sim}{n}}_i \cdot (\underset{\sim}{i}_1 - \underset{\sim}{i}_1') = 0 \qquad \text{Eq. 3.91}$$

$$\hat{\underset{\sim}{n}}_j \cdot (\underset{\sim}{j}_1 - \underset{\sim}{j}_1') = 0 \qquad \text{Eq. 3.92}$$

$$\hat{n}_k \cdot (k_1 - k_1') = 0 \qquad \text{Eq. 3.93}$$

The situation is similar to, but more general than, that originally described by McEntire [79]. As indicated in [79], $\hat{n}_i$, $\hat{n}_j$ and $\hat{n}_k$ can be found by solving the system of non-linear equations formed by Eqs 3.1 to 3.6 and Eqs 3.91 to 3.93. If no two vectors among $(i_1 - i_1')$, $(j_1 - j_1')$ and $(k_1 - k_1')$ are parallel to each other, this system will have a unique solution. Once such a solution is found (using Brown's algorithm or otherwise), [R] is directly obtainable from Eq. 3.0.

t can then be derived as shown in 3.3.2.3.

(ii) Second strategy : three contact points on P_i, two on P_j and one on P_k.

The advantage of this strategy over the previous one is that non-linear equations are avoided as a closed-form solution for [R] and t exists.

Let i_1, i_1', i_1'', j_1, j_1' and k_1 be the position vectors of the tip of the wand when it contacts P_i, P_j and P_k respectively. We now have $(i_1 - i_1')$ and $(i_1 - i_1'')$ parallel to P_i, and $(j_1 - j_1')$ parallel to P_j (Fig. 3.12b).

Hence

$$\hat{n}_i \cdot (i_1 - i_1') = 0 \qquad \text{Eq. 3.94}$$

$$\hat{n}_i \cdot (i_1 - i_1'') = 0 \qquad \text{Eq. 3.95}$$

$$\hat{n}_j \cdot (j_1 - j_1') = 0 \qquad \text{Eq. 3.96}$$

If $(i_1 - i_1')$ and $(i_1 - i_1'')$ are linearly independent (that is, if the three contact points on P_i are non-collinear), we can combine Eqs 3.1, 3.94 and 3.95 to give

$$\hat{n}_i = \pm \frac{(i_1 - i_1') \times (i_1 - i_1'')}{\left|(i_1 - i_1') \times (i_1 - i_1'')\right|}$$

The actual direction of $\hat{n}_i$ must be chosen such that $\hat{n}_i \cdot i_1$, $\hat{n}_i \cdot i_1'$, and $\hat{n}_i \cdot i_1''$ are negative.

Again, assuming that $\hat{n}_i$ and $(j_1 - j_1')$ are linearly independent we can

combine Eqs 3.2, 3.3 and 3.96 to give

$$\hat{\underset{\sim}{n}}_j = \pm \frac{\hat{\underset{\sim}{n}}_i \times (\underset{\sim}{j}_1 - \underset{\sim}{j}_1')}{|\hat{\underset{\sim}{n}}_i \times (\underset{\sim}{j}_1 - \underset{\sim}{j}_1')|}$$

As with $\hat{\underset{\sim}{n}}_i$, the actual direction of $\hat{\underset{\sim}{n}}_j$ must be chosen such that $\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{j}$ and $\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{j}'$ are negative. Once $\hat{\underset{\sim}{n}}_i$ and $\hat{\underset{\sim}{n}}_j$ are known, $\hat{\underset{\sim}{n}}_k$ can be obtained directly from Eqs 3.4 to 3.6. [R] is then given by Eq. 3.0 and $\underset{\sim}{t}$ can be derived as shown in 3.3.2.3.

(iii) Third strategy : three contact points on P_i, three on P_j, and one on P_k.

In strategy (ii), $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1')$ must not be parallel to $\hat{\underset{\sim}{n}}_i$ if $\hat{\underset{\sim}{n}}_j$ is to be determinable. This restriction is removed by introducing an additional contact point between P_j and the tip of the wand, while keeping the numbers of contact points on P_i and P_k the same as in (ii).

Let $\underset{\sim}{i}_1$, $\underset{\sim}{i}_1'$, $\underset{\sim}{i}_1''$, $\underset{\sim}{j}_1$, $\underset{\sim}{j}_1'$, $\underset{\sim}{j}_1''$ and $\underset{\sim}{k}_1$ be the position vectors of the tip of the wand as it contacts P_i, P_j and P_k respectively.

Again $(\underset{\sim}{i}_1 - \underset{\sim}{i}_1')$ and $(\underset{\sim}{i}_1 - \underset{\sim}{i}_1'')$ are parallel to P_i and $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1')$ is parallel to P_j. Consequently, Eqs 3.94 to 3.96 still apply. In addition, $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1'')$ is also parallel to P_j (see Fig. 3.12c) and hence

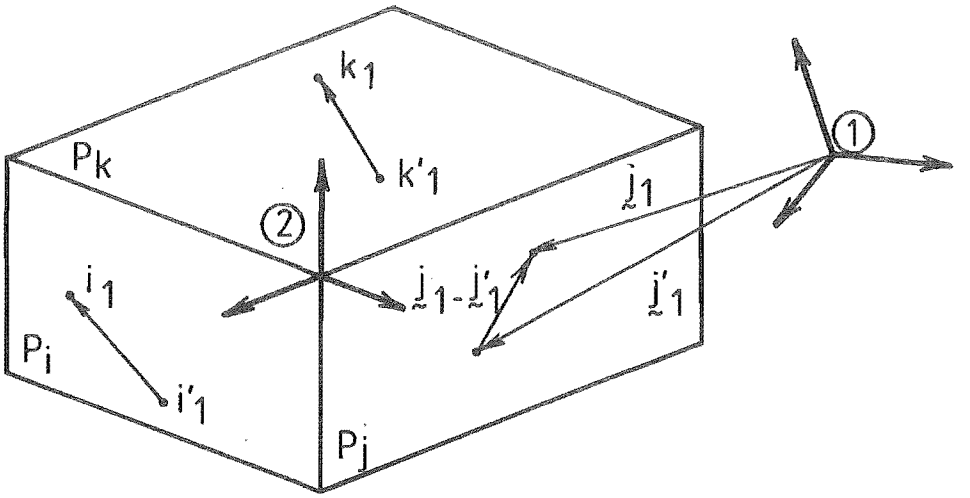
$$\hat{\underset{\sim}{n}}_j \cdot (\underset{\sim}{j}_1 - \underset{\sim}{j}_1'') = 0 \qquad \text{Eq. 3.97}$$

$\hat{\underset{\sim}{n}}_i$ is derived from Eqs 3.1, 3.94 and 3.95 exactly as in (ii), whereas, by virtue of Eqs 3.2, 3.96 and 3.97, $\hat{\underset{\sim}{n}}_j$ now becomes

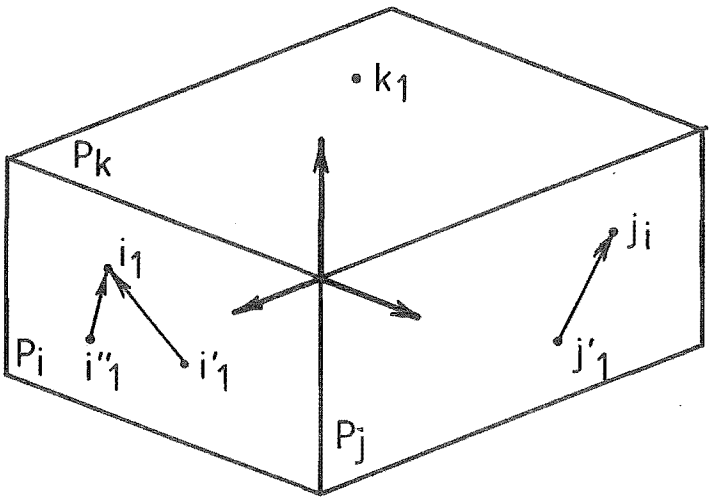
$$\hat{\underset{\sim}{n}}_j = \pm \frac{(\underset{\sim}{j}_1 - \underset{\sim}{j}_1') \times (\underset{\sim}{j}_1 - \underset{\sim}{j}_1'')}{|(\underset{\sim}{j}_1 - \underset{\sim}{j}_1') \times (\underset{\sim}{j}_1 - \underset{\sim}{j}_1'')|}$$

(assuming that $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1')$ and $(\underset{\sim}{j}_1 - \underset{\sim}{j}_1'')$ are linearly independent, or in other words, that the three contact points on P_j are non-collinear).

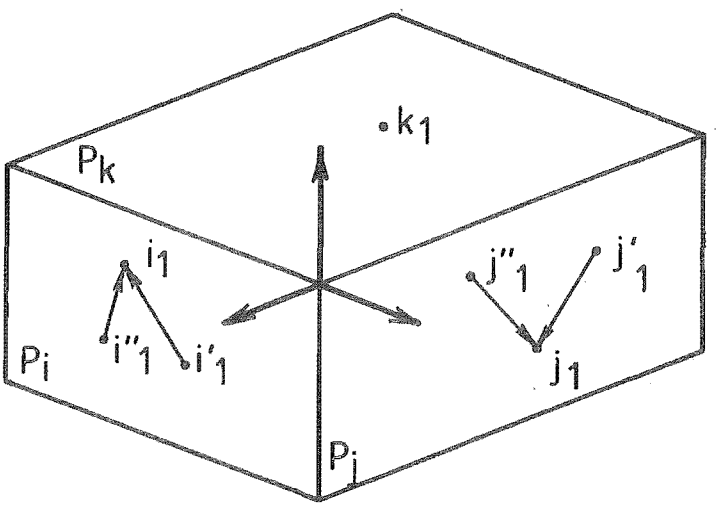
The actual direction of $\hat{\underset{\sim}{n}}_j$ again is dictated by the fact that $\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{j}_1$, $\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{j}_1'$ and $\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{j}_1''$ must be negative.



(a) first strategy



(b) second strategy



(c) third strategy

FIG 3·12 TOUCH-TOUCH SENSING

$\hat{n}_k$ is computed from $\hat{n}_i$ and $\hat{n}_j$ using Eqs 3.4 to 3.6. [R] is given by Eq. 3.0 and $\underset{\sim}{t}$ can be derived as shown in 3.3.2.3.

3.3.3 Discussion

Thus, it is theoretically possible to determine the misalignment between two contacting objects by comparing the observations of contact sensors associated with each object. Bilateral sensing, which operates on this comparison principle, is therefore a functional relation between sensor and error spaces. From our brief look at a family of simple bilateral sensing techniques, the following points have emerged:

(i) Bilateral sensing requires the use of computers. However, most of the numerical tasks involved are relatively simple and can be implemented in low-cost microcomputers.

(ii) Bilateral sensing relies heavily on position sensing. In force-force sensing, the displacement of the robot hand from the first to the second contact location must be measured. In all the other sensing schemes, the position of the tip of the wand relative to the robot hand must be found. However, note that neither the absolute location of the hand nor that of the work table need be known in any of the techniques considered. This suggests that the initial positioning of the hand with respect to the work table is unimportant and can therefore be performed quite inaccurately by a high-speed robot arm.

(iii) Bilateral sensing is a multinary relation between sensor and error spaces. Within the family of bilateral techniques described in 3.3.2, the number of sensory observations needed for uniquely specifying an error-space vector ranges between 2 (as in force-force sensing) and 7 (as in touch-touch sensing). The need for multiple observations and its implications will be discussed in Chapter 4.

3.4 SUMMARY

This chapter has been devoted to the study of feedback techniques. Both existing and new techniques have been examined from a theoretical viewpoint. The existing techniques have included (unilateral) contact sensing, visual sensing, and integrated sensing. The new techniques, known as bilateral contact sensing, are based on the use of two arrays of contact sensors. This combination has been shown to allow the spatial relationship between the two arrays to be found without the necessity for blind search or environmental structuring.

CHAPTER 4

BILATERAL FEEDBACK : A SPECIAL CASE AND A GENERAL THEOREM

Force-force sensing, a special case of bilateral sensing already outlined in Chapter 3, is re-examined in more detail in the present chapter. First, a closed-form solution is derived, which will yield the relative location of two force-sensing arrays when two pairs of force-moment vectors measured in one array are matched with two corresponding pairs in the other array. The necessity for making at least two pairs of measurements in each array is proven for both the case of small misalignments and the case of arbitrary misalignments between the two arrays. The approach adopted in the proof for the second case leads to a convenient kinematic representation of force-force sensing, which is then used to establish a theorem on bilateral sensing in general.

4.1 FORCE-FORCE SENSING

It is recalled that, if $(\underset{\sim}{f}_1, \underset{\sim}{m}_1)$ and $(\underset{\sim}{f}_2, \underset{\sim}{m}_2)$ are the forces and moments detected by two force-sensing arrays A_1 and A_2 in response to internal forces $\underset{\sim}{f}$ and $-\underset{\sim}{f}$ between A_1 and A_2, then for equilibrium (Fig.4.1)

$$\underset{\sim}{f}_1 = -[R]\underset{\sim}{f}_2 \qquad \text{Eq. 4.1}$$

$$\underset{\sim}{m}_1 + [R]\underset{\sim}{m}_2 = \underset{\sim}{t} \times \underset{\sim}{f}_1 \qquad \text{Eq. 4.2}$$

where [R] is a 3x3 orthogonal matrix describing the orientation of A_2 relative to A_1 and $\underset{\sim}{t}$ is a 3x1 vector representing the position of A_2 relative to A_1.

Similarly, when the internal forces are changed to $\underset{\sim}{F}$ and $-\underset{\sim}{F}$ and the detected forces and moments to $(\underset{\sim}{F}_1, \underset{\sim}{M}_1)$ and $(\underset{\sim}{F}_2, \underset{\sim}{M}_2)$ without altering the relative orientation [R] and relative position $\underset{\sim}{t}$ of A_1 and A_2,

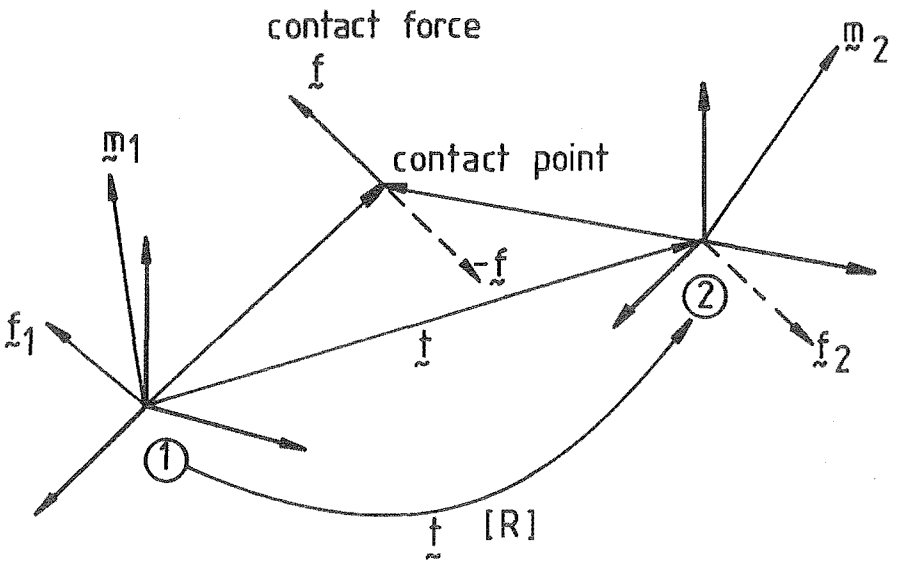


FIG 4·1 FORCE-FORCE SENSING

$$\underset{\sim}{F}_1 = -[R]\underset{\sim}{F}_2 \qquad \text{Eq. 4.3}$$

$$\underset{\sim}{M}_1 + [R]\underset{\sim}{M}_2 = \underset{\sim}{t} \times \underset{\sim}{F}_1 \qquad \text{Eq. 4.4}$$

A numerical procedure for obtaining [R] and $\underset{\sim}{t}$ from the above equations and from the orthogonality conditions of [R] was suggested in Chapter 3. We shall now determine [R] and $\underset{\sim}{t}$ vectorially instead. The determination of [R] is based on a formula first published by Rodrigues in 1840 in connection with spatial displacements of rigid bodies [80].

4.1.1 The Rotation Matrix [R]

Let us rewrite Eq. 4.1 as

$$\underset{\sim}{f}_1 = [R](-\underset{\sim}{f}_2) \qquad \text{Eq. 4.5}$$

which suggests that $\underset{\sim}{f}_1$ can be regarded as the vector $(-\underset{\sim}{f}_2)$ after the rigid-body rotation [R] about an axis $\hat{\underset{\sim}{a}}$ and through an angle δ.

Consequently, we can apply the vectorial version of Rodrigues formula relating $\underset{\sim}{f}_1$, $(-\underset{\sim}{f}_2)$, $\hat{\underset{\sim}{n}}$ and δ [81], viz

$$\sin(\tfrac{\delta}{2})\,\hat{\underset{\sim}{a}} \times (\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = \cos(\tfrac{\delta}{2})\,(\underset{\sim}{f}_1 + \underset{\sim}{f}_2) \qquad \text{Eq. 4.6}$$

Similarly, from Eq. 4.3 we can write

$$\sin(\tfrac{\delta}{2})\,\hat{\underset{\sim}{a}} \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = \cos(\tfrac{\delta}{2})\,(\underset{\sim}{F}_1 + \underset{\sim}{F}_2) \qquad \text{Eq. 4.7}$$

First, using Eqs 4.6 and 4.7, we shall find $\hat{\underset{\sim}{a}}$ and δ in terms of $\underset{\sim}{f}_1$, $\underset{\sim}{f}_2$, $\underset{\sim}{F}_1$, and $\underset{\sim}{F}_2$. Once $\hat{\underset{\sim}{a}}$ and δ are known, we can obtain [R] by equating it to the Euler matrix [78].

$$\begin{bmatrix} \ell^2 + (1-\ell^2)\cos\delta & \ell m\,(1-\cos\delta) - n\sin\delta & \ell n\,(1-\cos\delta) + m\sin\delta \\ m\ell\,(1-\cos\delta) + n\sin\delta & m^2 + (1-m^2)\cos\delta & mn\,(1-\cos\delta) - \ell\sin\delta \\ n\ell\,(1-\cos\delta) - m\sin\delta & nm\,(1-\cos\delta) + \ell\sin\delta & n^2 + (1-n^2)\cos\delta \end{bmatrix}$$

where ℓ, m, and n are the components of the unit vector $\hat{\underset{\sim}{a}}$.

We shall distinguish two possibilities :

1. $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0$

2. $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) \neq 0$

4.1.1.1 First possibility: $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0$

(i) If $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0$ (Fig. 4.2a) then from Eqs 4.1 and 4.3

$$[R]\underset{\sim}{f}_2 + \underset{\sim}{f}_2 = [R]\underset{\sim}{F}_2 + \underset{\sim}{F}_2 = 0 \qquad \text{Eq.4.8}$$

As $\underset{\sim}{f}_2$ and $\underset{\sim}{F}_2$ are non zero, Eq. 4.8 implies that $\varepsilon = -1$ is an eigenvalue of [R]. From Thompson [82], ε is related to δ by

$$\varepsilon = e^{i\delta}$$

where $$i^2 \equiv -1$$

$$\therefore \quad \delta = \Pi$$

Also, since [R] is orthogonal

$$[R](\underset{\sim}{f}_2 \times \underset{\sim}{F}_2) = [R]\underset{\sim}{f}_2 \times [R]\underset{\sim}{F}_2 \qquad \text{Eq. 4.9}$$

Combining Eqs 4.8 and 4.9 gives

$$[R](\underset{\sim}{f}_2 \times \underset{\sim}{F}_2) = \underset{\sim}{f}_2 \times \underset{\sim}{F}_2 \qquad \text{Eq. 4.10}$$

If $\underset{\sim}{f}_2$ and $\underset{\sim}{F}_2$ are linearly independent, Eq. 4.10 means that $\underset{\sim}{f}_2 \times \underset{\sim}{F}_2$ is an invariant vector of the rotation [R]. That is, $\underset{\sim}{f}_2 \times \underset{\sim}{F}_2$ and $\hat{\underset{\sim}{a}}$ are parallel, or

$$\hat{\underset{\sim}{a}} = \frac{\underset{\sim}{f}_2 \times \underset{\sim}{F}_2}{\left|\underset{\sim}{f}_2 \times \underset{\sim}{F}_2\right|}$$

(ii) If $\underset{\sim}{f}_1 - \underset{\sim}{f}_2 \neq 0$

and (a) $\underset{\sim}{F}_1 - \underset{\sim}{F}_2 = 0$ (Fig. 4.2b)

then $\delta = \Pi$ as in case (i).

To find $\hat{\underset{\sim}{a}}$, note that $[R](\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = [R]\underset{\sim}{f}_1 - [R]\underset{\sim}{f}_2$

$$= -[R][R]\underset{\sim}{f}_2 + \underset{\sim}{f}_1 \qquad \text{Eq. 4.11}$$

Since [R] is a rotation through Π, [R][R] is the identity matrix and Eq. 4.11 becomes

$$[R](\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = \underset{\sim}{f}_1 - \underset{\sim}{f}_2 \qquad \text{Eq. 4.12}$$

which again means $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2)$ is parallel to $\hat{\underset{\sim}{a}}$ and

$$\hat{\underset{\sim}{a}} = \frac{\underset{\sim}{f}_1 - \underset{\sim}{f}_2}{|\underset{\sim}{f}_1 - \underset{\sim}{f}_2|}$$

or (b) $\underset{\sim}{F}_1 - \underset{\sim}{F}_2 \neq 0$ (Fig. 4.2c)

then $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0$ implies

$$(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = \zeta(\underset{\sim}{F}_1 - \underset{\sim}{F}_2) \qquad \text{Eq. 4.13}$$

where ζ is a non-zero scalar.

Let $\underset{\sim}{p}_1 = \underset{\sim}{f}_1 - \zeta\underset{\sim}{F}_1$ Eq. 4.14

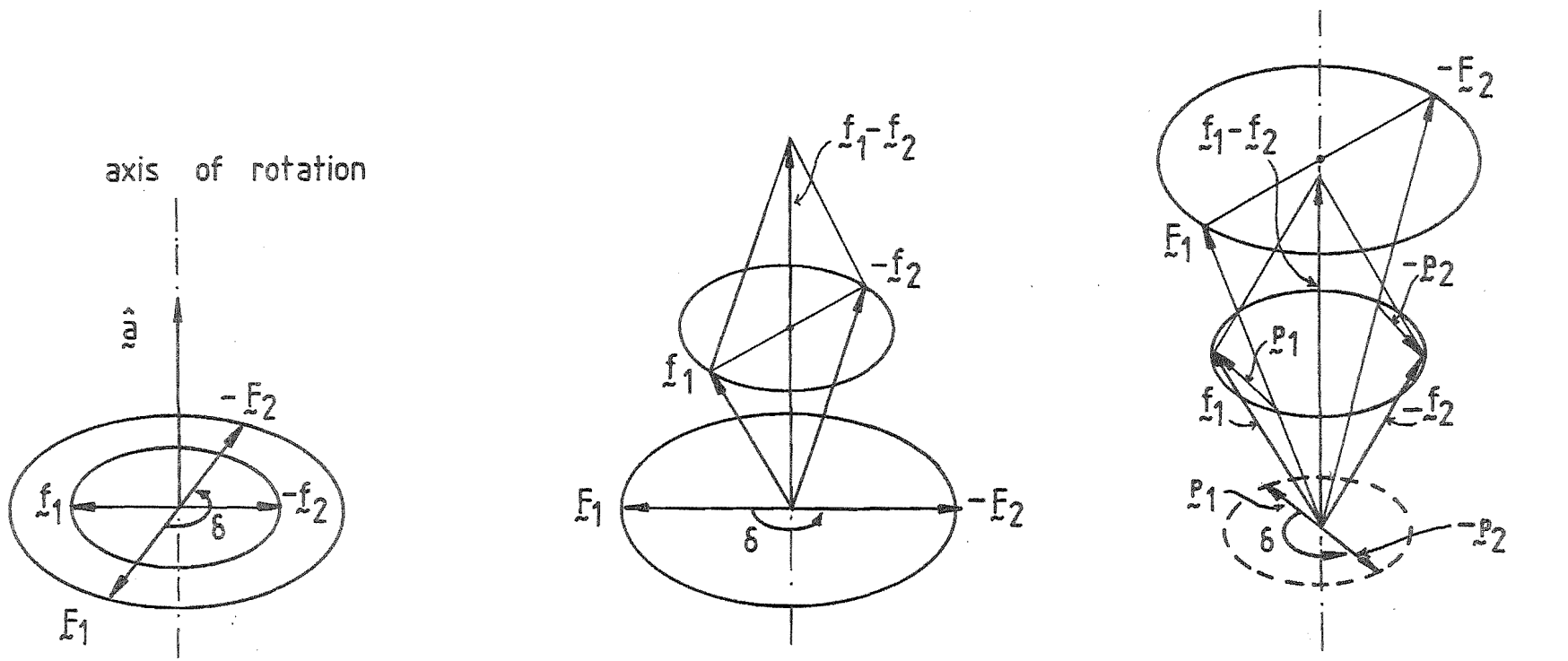
and $\underset{\sim}{p}_2 = \underset{\sim}{f}_2 - \zeta\underset{\sim}{F}_2$ Eq. 4.15

Eq. 4.13 can then be rewritten as

$$\underset{\sim}{p}_1 - \underset{\sim}{p}_2 = 0 \qquad \text{Eq. 4.16}$$

From Eqs 4.8, 4.14 and 4.15

$$\underset{\sim}{p}_1 = -[R]\underset{\sim}{p}_2 \qquad \text{Eq. 4.17}$$



(a) $\underset{\sim}{f}_1 - \underset{\sim}{f}_2 = \underset{\sim}{F}_1 - \underset{\sim}{F}_2 = 0$

(b) $\underset{\sim}{f}_1 - \underset{\sim}{f}_2 \neq 0 \;;\; \underset{\sim}{F}_1 - \underset{\sim}{F}_2 = 0$

(c) $\underset{\sim}{f}_1 - \underset{\sim}{f}_2 \neq 0 \;;\; \underset{\sim}{F}_1 - \underset{\sim}{F}_2 \neq 0$

FIG 4.2 AXIS AND ANGLE OF ROTATION WHEN $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0$

As proved in case (i), Eqs 4.16 and 4.17 yield $\delta = \Pi$.

$\hat{\underset{\sim}{a}}$ can be derived as shown in case (ii)(a).

Hence $$\hat{\underset{\sim}{a}} = \frac{\underset{\sim}{f}_1 - \underset{\sim}{f}_2}{|\underset{\sim}{f}_1 - \underset{\sim}{f}_2|} = \frac{\underset{\sim}{F}_1 - \underset{\sim}{F}_2}{|\underset{\sim}{F}_1 - \underset{\sim}{F}_2|}$$

4.1.1.2 Second possibility: $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) \neq 0$

First we shall prove that, unlike in the previous section, δ cannot be equal to Π.

Suppose $\delta = \Pi$.

Then Eqs 4.6 and 4.7 simplify to

$$\hat{\underset{\sim}{a}} \times (\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = 0 \qquad \text{Eq. 4.18}$$

$$\hat{\underset{\sim}{a}} \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = 0 \qquad \text{Eq. 4.19}$$

Since $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \neq 0$ (unless the condition $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) \neq 0$ is to be violated), Eq. 4.18 yields

$$\hat{\underset{\sim}{a}} = \frac{\underset{\sim}{f}_1 - \underset{\sim}{f}_2}{|\underset{\sim}{f}_1 - \underset{\sim}{f}_2|} \qquad \text{Eq. 4.20}$$

However, substituting Eq. 4.20 in Eq. 4.19 gives

$$\frac{(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2)}{|\underset{\sim}{f}_1 - \underset{\sim}{f}_2|} = 0, \quad \text{which is impossible by hypothesis.}$$

Hence, $\delta \neq \Pi$ for all cases where $(\underset{\sim}{f}_1 - \underset{\sim}{f}_2) \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) \neq 0$; and we can rewrite Eqs 4.6 and 4.7 as

$$\tan(\frac{\delta}{2})\ \hat{\underset{\sim}{a}} \times (\underset{\sim}{f}_1 - \underset{\sim}{f}_2) = \underset{\sim}{f}_1 + \underset{\sim}{f}_2 \qquad \text{Eq. 4.21}$$

$$\tan(\frac{\delta}{2})\ \hat{\underset{\sim}{a}} \times (\underset{\sim}{F}_1 - \underset{\sim}{F}_2) = \underset{\sim}{F}_1 + \underset{\sim}{F}_2 \qquad \text{Eq. 4.22}$$

For convenience, let

$$\underset{\sim}{w} = \tan\left(\frac{\delta}{2}\right) \hat{\underset{\sim}{a}} \qquad \text{Eq. 4.23}$$

$$\underset{\sim}{x}_1 = \underset{\sim}{f}_1 - \underset{\sim}{f}_2$$

$$\underset{\sim}{y}_1 = \underset{\sim}{f}_1 + \underset{\sim}{f}_2$$

$$\underset{\sim}{x}_2 = \underset{\sim}{F}_1 - \underset{\sim}{F}_2$$

$$\underset{\sim}{y}_2 = \underset{\sim}{F}_1 + \underset{\sim}{F}_2$$

and Eqs 4.21 and 4.22 become

$$\underset{\sim}{w} \times \underset{\sim}{x}_1 = \underset{\sim}{y}_1 \qquad \text{Eq. 4.24}$$

$$\underset{\sim}{w} \times \underset{\sim}{x}_2 = \underset{\sim}{y}_2 \qquad \text{Eq. 4.25}$$

Therefore

$$(\underset{\sim}{w} \times \underset{\sim}{x}_1) \times (\underset{\sim}{w} \times \underset{\sim}{x}_2) = \underset{\sim}{y}_1 \times \underset{\sim}{y}_2$$

which, upon development and simplification, gives

$$\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2 \, \underset{\sim}{w} = \underset{\sim}{y}_1 \times \underset{\sim}{y}_2 \qquad \text{Eq. 4.26}$$

(i) If $\underset{\sim}{y}_1 \times \underset{\sim}{y}_2 \neq 0$ (Fig. 4.3a)

Then from Eq. 4.26, $\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2 \neq 0$ and

$$\underset{\sim}{w} = \frac{\underset{\sim}{y}_1 \times \underset{\sim}{y}_2}{\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2} \qquad \text{Eq. 4.27}$$

Using Eqs 4.23 and 4.27, we deduce that

$$\delta = 2 \tan^{-1} \left| \frac{\underset{\sim}{y}_1 \times \underset{\sim}{y}_2}{\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2} \right|$$

$$\text{and } \hat{\underset{\sim}{a}} = \left(\frac{1}{\tan\left(\frac{\delta}{2}\right)} \right) \frac{\underset{\sim}{y}_1 \times \underset{\sim}{y}_2}{\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2}$$

(ii) If $\underset{\sim}{y}_1 \times \underset{\sim}{y}_2 = 0$ (Fig. 4.3b)

Then due to Eq. 4.26

$$\underset{\sim}{y}_1 \cdot \underset{\sim}{x}_2 = 0 \qquad \text{Eq. 4.28}$$

Combining Eqs 4.24 and 4.28 gives

$$(\underset{\sim}{w} \times \underset{\sim}{x}_1) \cdot \underset{\sim}{x}_2 = 0$$

Hence $\underset{\sim}{w} = \alpha \underset{\sim}{x}_1 + \beta \underset{\sim}{x}_2$ where α and β are scalars.

Substituting $\underset{\sim}{w}$ in Eqs 4.24 and 4.25 gives

$$\beta(\underset{\sim}{x}_2 \times \underset{\sim}{x}_1) = \underset{\sim}{y}_1$$

$$\alpha(\underset{\sim}{x}_1 \times \underset{\sim}{x}_2) = \underset{\sim}{y}_2$$

$$\therefore \underset{\sim}{w} = \frac{\underset{\sim}{y}_2 \cdot (\underset{\sim}{x}_1 \times \underset{\sim}{x}_2)\underset{\sim}{x}_1 - \underset{\sim}{y}_1 \cdot (\underset{\sim}{x}_1 \times \underset{\sim}{x}_2)\underset{\sim}{x}_2}{(\underset{\sim}{x}_1 \times \underset{\sim}{x}_2)^2}$$

δ and $\underset{\sim}{\hat{a}}$ can be deduced from $\underset{\sim}{w}$ as in (i).

NOTE: When $\underset{\sim}{y}_1 = \underset{\sim}{y}_2 = 0$

$$\underset{\sim}{w} = 0$$

$\delta = 0$ and $\underset{\sim}{\hat{a}}$ is indeterminate

$\therefore$ [R] = [I], the identity matrix.

4.1.2 The Translation Vector $\underset{\sim}{t}$

Once the rotation matrix [R] is determined, the L.H.S. of Eq.4.2 is known and can be grouped into a single vector $\underset{\sim}{g}_1$

$$\underset{\sim}{g}_1 = \underset{\sim}{t} \times \underset{\sim}{f}_1 \qquad \text{Eq. 4.29}$$

Similarly, the L.H.S. of Eq. 4.4 can be grouped into a single vector $\underset{\sim}{G}_1$

$$\underset{\sim}{G}_1 = \underset{\sim}{t} \times \underset{\sim}{F}_1 \qquad \text{Eq. 4.30}$$

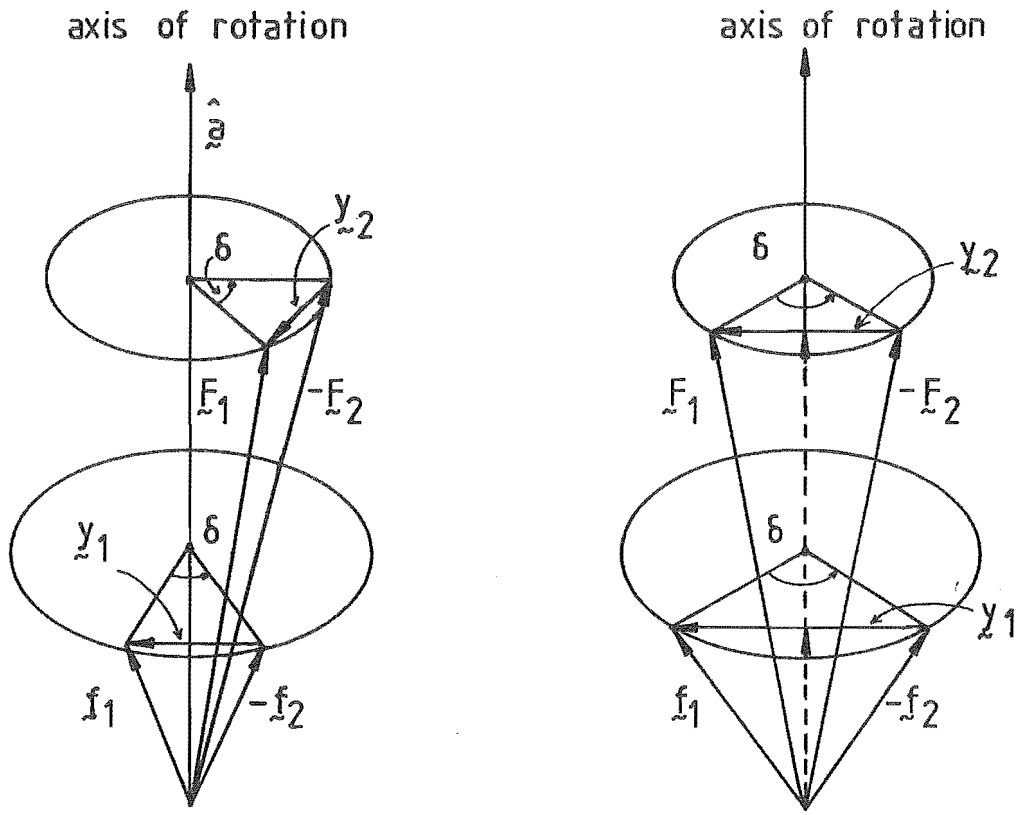


FIG 4·3 AXIS AND ANGLE OF ROTATION WHEN $(f_1 - f_2) \times (F_1 - F_2) \neq 0$

Eqs. 4.29 and 4.30 are identical in form to Eqs 4.24 and 4.25. Therefore, as was shown for $\underset{\sim}{w}$ in section 4.1.1.2

$$\underset{\sim}{t} = \frac{\underset{\sim}{g}_1 \times \underset{\sim}{G}_1}{\underset{\sim}{g}_1 \cdot \underset{\sim}{F}_1} \qquad \text{when } \underset{\sim}{g}_1 \times \underset{\sim}{G}_1 \neq 0$$

or

$$\underset{\sim}{t} = \frac{\underset{\sim}{G}_1 \cdot (\underset{\sim}{f}_1 \times \underset{\sim}{F}_1)\underset{\sim}{f}_1 - \underset{\sim}{g}_1 \cdot (\underset{\sim}{f}_1 \times \underset{\sim}{F}_1)\underset{\sim}{F}_1}{(\underset{\sim}{f}_1 \times \underset{\sim}{F}_1)^2} \qquad \text{when } \underset{\sim}{g}_1 \times \underset{\sim}{G}_1 = 0$$

4.1.3 Discussion

In sections 4.1.1 and 4.1.2, we have determined, or shown how to determine, the relative location [R] and $\underset{\sim}{t}$ of two force-sensing arrays A_1 and A_2, for an exhaustive combination of forces and moments mutually experienced by these arrays.

The need to measure more than one pair of force-moment vectors in each array should have been obvious from the process of deriving [R] and $\underset{\sim}{t}$. However, we shall provide two additional proofs which will conclusively establish this need.

The first proof applies when the angular misalignment between A_1 and A_2 is small and the second proof, when A_1 and A_2 are arbitrarily located. In both cases, we shall demonstrate that [R] and $\underset{\sim}{t}$ cannot be uniquely specified if one pair of measurements only is used.

4.1.3.1 Small misalignments.

For a small angular misalignment, [R] can be thought of as representing a rotation through a small angle δ about some axis $\underset{\sim}{\hat{a}}$, and can therefore be linearised as

$$[R] = \begin{bmatrix} 1 & -n\delta & m\delta \\ n\delta & 1 & -\ell\delta \\ -m\delta & \ell\delta & 1 \end{bmatrix}$$

where ℓ, m, and n are again the components of the unit vector $\underset{\sim}{\hat{a}}$.

Let $\underset{\sim}{f}_1 = \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{z1} \end{bmatrix}$ and $\underset{\sim}{f}_2 = \begin{bmatrix} f_{x2} \\ f_{y2} \\ f_{z2} \end{bmatrix}$ be the contact

forces sensed by A_1 and A_2 respectively.

Eq. 4.1 then becomes

$$\begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{z1} \end{bmatrix} = - \begin{bmatrix} 1 & -n\delta & m\delta \\ n\delta & 1 & -\ell\delta \\ -m\delta & \ell\delta & 1 \end{bmatrix} \begin{bmatrix} f_{x2} \\ f_{y2} \\ f_{z2} \end{bmatrix} \quad \text{Eq. 4.31}$$

which, upon rearrangement gives

$$[A] \underset{\sim}{w}^* = \underset{\sim}{b} \quad \text{Eq. 4.32}$$

where $[A] = \begin{bmatrix} 0 & -f_{z2} & f_{y2} \\ f_{z2} & 0 & -f_{x2} \\ -f_{y2} & f_{x2} & 0 \end{bmatrix}$

$$\underset{\sim}{w}^* = \begin{bmatrix} \ell\delta \\ m\delta \\ n\delta \end{bmatrix}$$

and $\underset{\sim}{b} = \begin{bmatrix} f_{x1} + f_{x2} \\ f_{y1} + f_{y2} \\ f_{z1} + f_{z2} \end{bmatrix}$

Clearly, [A] is a singular matrix, Eq. 4.32 has an infinite number of solutions $\underset{\sim}{w}^*$, and therefore [R] cannot be uniquely determined.

The indeterminacy of $\underset{\sim}{t}$ follows directly from Eq. 4.2 which can be rewritten as

$$[B] \underset{\sim}{t} = \underset{\sim}{g}_1 \quad \text{Eq. 4.33}$$

where $[B] = \begin{bmatrix} 0 & f_{zl} & -f_{yl} \\ -f_{zl} & 0 & f_{xl} \\ f_{yl} & -f_{xl} & 0 \end{bmatrix}$

and $\underset{\sim}{g}_1 = \underset{\sim}{m}_1 + [R]\underset{\sim}{m}_2$

Again, it is readily appreciated that even if $\underset{\sim}{g}_1$ were unique, $\underset{\sim}{t}$ would not be so, as [B] is a singular matrix.

4.1.3.2 Arbitrary misalignments

As before, let $\underset{\sim}{f}_1$ and $\underset{\sim}{f}_2$ be the forces sensed by A_1 and A_2 respectively, and rewrite Eq. 4.1 as

$$[R]^T \underset{\sim}{f}_1 = -\underset{\sim}{f}_2$$

or $$\underset{\sim}{f}_1^T [R] = -\underset{\sim}{f}_2^T \qquad \text{Eq. 4.34}$$

where $[R]^T$, $\underset{\sim}{f}_1^T$ and $\underset{\sim}{f}_2^T$ are the transpose of [R], $\underset{\sim}{f}_1$ and $\underset{\sim}{f}_2$ respectively.

Substituting $[R] = [\hat{\underset{\sim}{n}}_i \; \hat{\underset{\sim}{n}}_j \; \hat{\underset{\sim}{n}}_k]$ in Eq. 4.34 yields

$$\hat{\underset{\sim}{n}}_i \cdot \underset{\sim}{f}_1 = -f_{x2} \qquad \text{Eq. 4.35}$$

$$\hat{\underset{\sim}{n}}_j \cdot \underset{\sim}{f}_1 = -f_{y2} \qquad \text{Eq. 4.36}$$

$$\hat{\underset{\sim}{n}}_k \cdot \underset{\sim}{f}_1 = -f_{z2} \qquad \text{Eq. 4.37}$$

f_{x2}, f_{y2} and f_{z2} being the components of $\underset{\sim}{f}_z$.

Equations 4.35 to 4.37 specify the projections along a given vector $\underset{\sim}{f}_1$, of the triad of unit direction vectors $(\hat{\underset{\sim}{n}}_i, \hat{\underset{\sim}{n}}_j, \hat{\underset{\sim}{n}}_k)$. Suppose $(\hat{\underset{\sim}{n}}_i^*, \hat{\underset{\sim}{n}}_j^*, \hat{\underset{\sim}{n}}_k^*)$ is a triad which projects onto $\underset{\sim}{f}_1$ according to these equations. It is obvious that if this triad is spun about $\underset{\sim}{f}_1$ to some new orientation, the new triad $(\hat{\underset{\sim}{n}}_i^*, \hat{\underset{\sim}{n}}_j^*, \hat{\underset{\sim}{n}}_k^*)_{NEW}$ thus obtained will also satisfy Eqs 4.35 to 4.37 (Fig. 4.4). In fact, there will be an infinite number of triads $(\hat{\underset{\sim}{n}}_i, \hat{\underset{\sim}{n}}_j, \hat{\underset{\sim}{n}}_k)$, and hence an infinite number of matrices [R], corresponding to any pair of force vectors $\underset{\sim}{f}_1$ and $\underset{\sim}{f}_2$.

The indeterminacy of $\underset{\sim}{t}$ can be proven algebraically with the help of Eq. 4.2, as in Section 4.1.3.1. Geometrically, however, Eq. 4.2 specifies the moment $\underset{\sim}{g}_1$, about the origin O_1 of array A_1, due to the direction vector $\underset{\sim}{f}_1$, of a straight line l through the origin O_2 of array A_2; $\underset{\sim}{t}$, the position vector of O_2 relative to O_1 is the moment arm of $\underset{\sim}{g}_1$. Fig. 4.5 shows array A_2 at some position relative to A_1, the position vector $\underset{\sim}{t}^*$ of O_2 satisfying Eq.4.2. Clearly, if A_2 is slid along l to a new position, the new vector $\underset{\sim NEW}{t^*}$ thus obtained will also satisfy Eq.4.2. In fact, there will be an infinite number of possible positions for A_2, and hence of vectors $\underset{\sim}{t}$, corresponding to any pair of force and moment vectors $\underset{\sim}{f}_1$ and $\underset{\sim}{g}_1$.

The existence of rotational and translational freedom between A_1 and A_2 when feedback information is underspecified suggests the kinematic model of Fig.4.6a. As shown in this figure, l is the line of action of the contact force between A_1 and A_2 (the force that manifests itself as $\underset{\sim}{f}_1$, $\underset{\sim}{m}_1$ in A_1 and $\underset{\sim}{f}_2$, $\underset{\sim}{m}_2$ in A_2). The triads of coordinate axes associated with A_1 and A_2 are joined to l by cylinder pairs C_1 and C_2 which allow these triads to rotate about, and slide along, l relatively to each other. It is easy to visualize that these motions do not affect how A_1 and A_2 perceive the contact force, i.e., as A_1 and A_2 are displaced, they still 'see' the force as ($\underset{\sim}{f}_1$, $\underset{\sim}{m}_1$) and ($\underset{\sim}{f}_2$, $\underset{\sim}{m}_2$) respectively. Therefore, conversely we can deduce that there exists more than one relative location between A_1 and A_2 for which a given contact force is 'seen' as ($\underset{\sim}{f}_1$, $\underset{\sim}{m}_1$) in A_1, and ($\underset{\sim}{f}_2$, $\underset{\sim}{m}_2$) in A_2.

The locking effect of an extra contact force between A_1 and A_2 is illustrated in Fig.4.6b (for simplicity, we have shown A_1 as rigidly linked to the lines of action of the two contact forces). That A_1 and A_2 can no longer move relatively to each other can be ascertained by computing the number of degrees of freedom F of the linkage shown in the above Figure.

For a spatial linkage with no general constraints, F is given by Kutzbach formula [83]

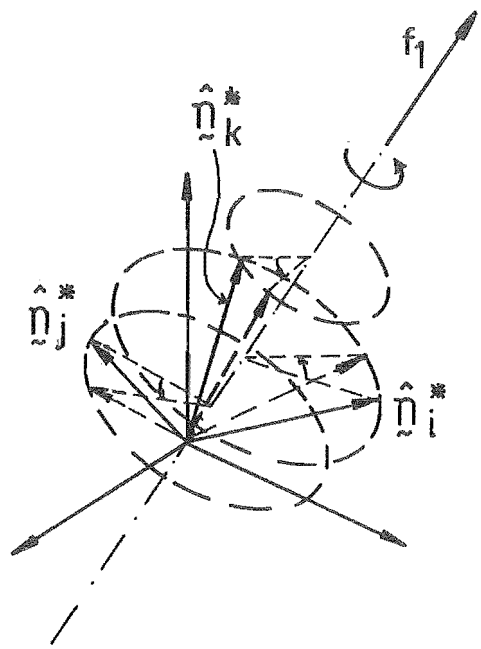


FIG. 4.4 AMBIGUOUS ORIENTATION

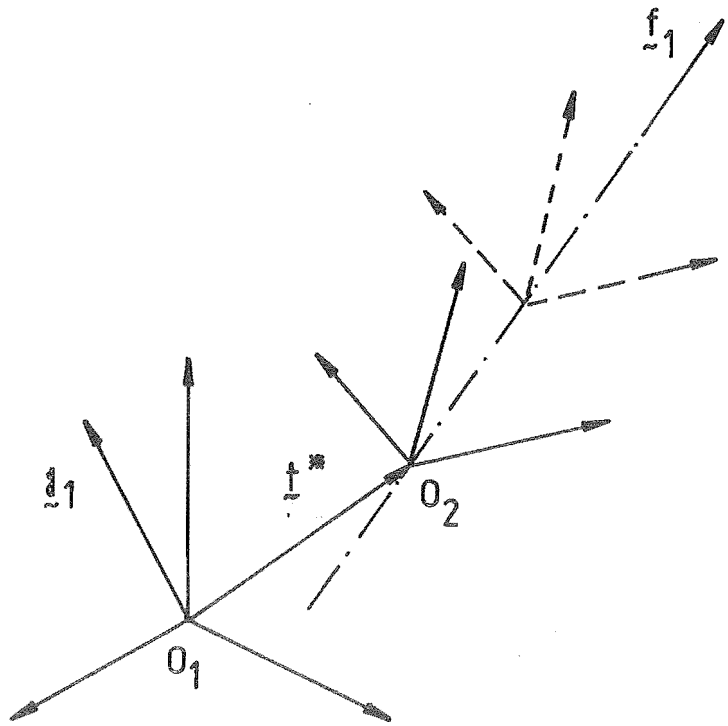


FIG 4.5 AMBIGUOUS POSITION

$$F = 6\ (n - 1)\ - 6j + \Sigma f \qquad \text{Eq. 4.38}$$

where n = number of links (including the fixed link)

j = number of joints (or pairs)

Σf = total number of degrees of freedom in all the joints.

In the linkage of Fig.4.6b,

$n = 2$

$j = 2$

and $\Sigma f = 4$.

$\therefore\ F = -2$.

As $F \leqslant 0$, the linkage is a structure and the relative location between A_1 and A_2 is completely specified.

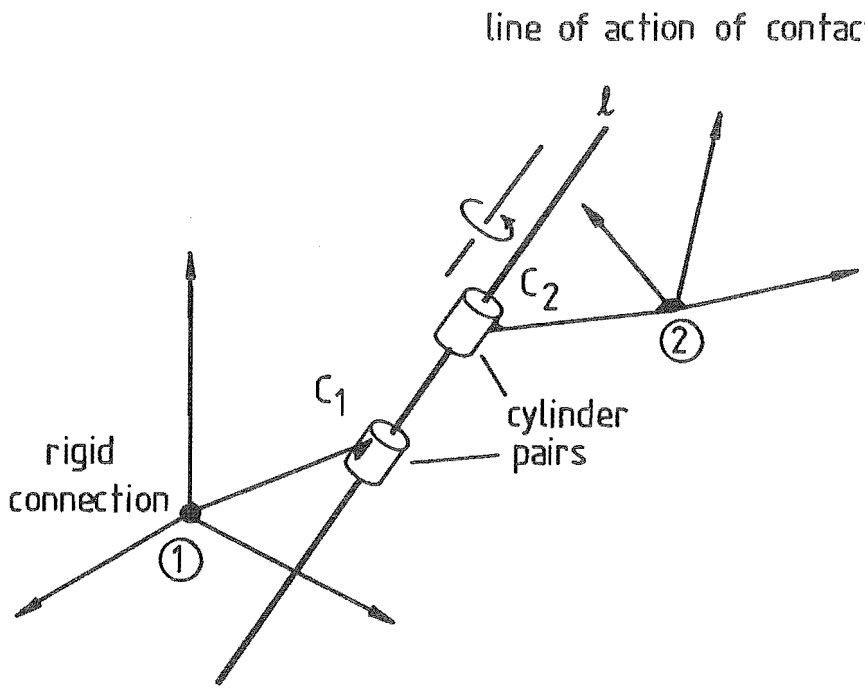
4.2 BILATERAL SENSING : A KINEMATIC PERSPECTIVE

Thus, using a simple kinematic model has enabled us, without having recourse to mathematics, to demonstrate the multinary character of force-force sensing, a member of the family of bilateral techniques outlined in Chapter 3. We shall now extend kinematic modelling to the remaining members of this family, namely touch-force, force-touch, and touch-touch sensing. We shall first construct kinematic models to show that these particular techniques, also, are multinary and then generalise our arguments to prove the multinary nature inherent in all bilateral techniques.

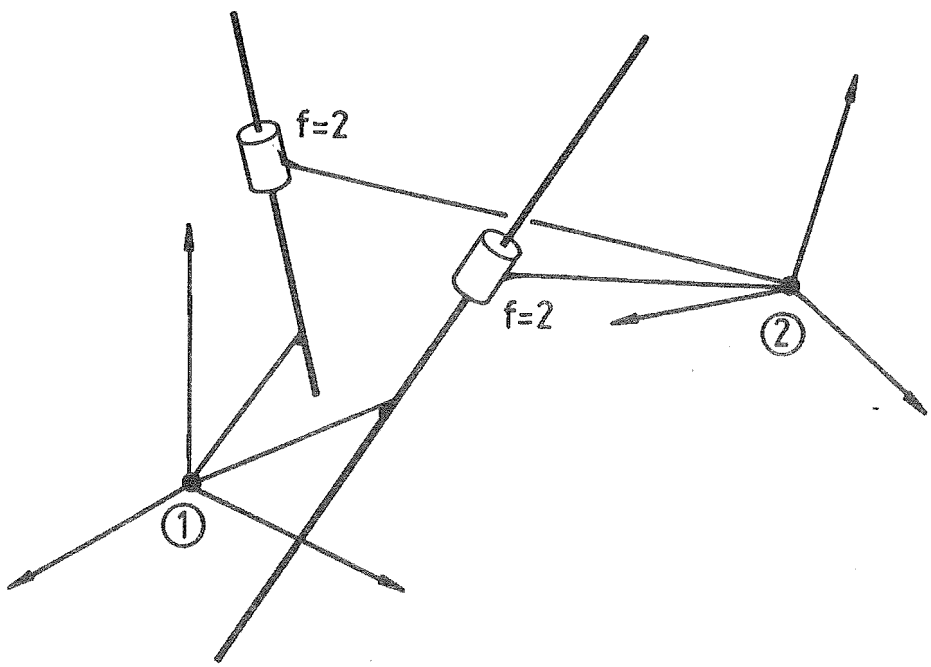
4.2.1 Kinematic Models

4.2.1.1 Touch-force sensing

We may recall that A_1 is now a wand which can detect touching at its sensitized tip, and A_2 is still an array of force sensors capable of measuring forces and moments in three dimensions. If the tip of the wand contacts an object B_2 fixed to A_2, then the position of the contact point is known relatively to the coordinate system of A_2 and the location of the line of



(a) RELATIVE FREEDOM BETWEEN TWO FORCE-SENSING ARRAYS



(b) LOCKING BETWEEN TWO FORCE-SENSING ARRAYS

FIG 4·6 KINEMATIC MODELS OF FORCE-FORCE SENSING

action of the contact force applied by the wand on B_2 is known relatively to the coordinate system of A_2.

Fig. 4.7a depicts this situation, showing the position vector $\underset{\sim}{U}_1$ of a contact point U, fixed to the coordinate system of A_1 and the line of action ℓ_u of a contact force applied at U, fixed to the coordinate system of A_2.

As indicated in Section 3.3.2.2, we can intersect ℓ_u with the boundary S of B_2 and obtain the position of U relative to A_2. Let this position be represented by vector $\underset{\sim}{U}_2$ fixed to the coordinate system of A_2. Obviously, $\underset{\sim}{U}_1$ and $\underset{\sim}{U}_2$ must meet at U, and A_1 and A_2 are free to revolve about any axis through U without affecting $\underset{\sim}{U}_1$ and $\underset{\sim}{U}_2$. This is modelled in Fig. 4.7b by joining A_1 and A_2 with a spheric pair at the contact point U.

Likewise, when the positions of two additional contact points V and W are known relatively to both A_1 and A_2, we can join A_1 and A_2 with spheric pairs at V and W. The resulting linkage is shown in Fig. 4.7c.

For this linkage,

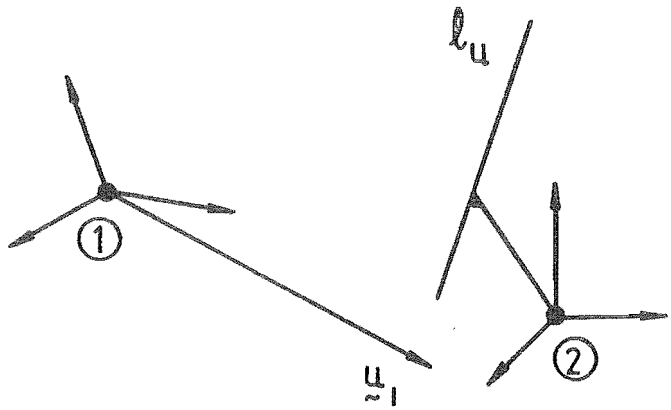
$$n = 2$$

$$j = 3$$

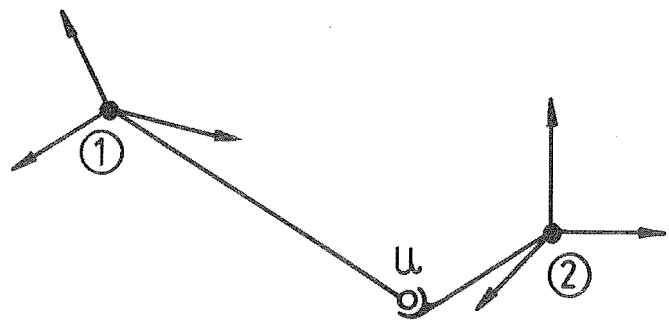
$$\text{and } \Sigma f = 9.$$

Applying Kutzbach formula (Eq. 4.38) we deduce that F = 3, i.e., the linkage is a structure and the relative location between A_1 and A_2 is completely specified.

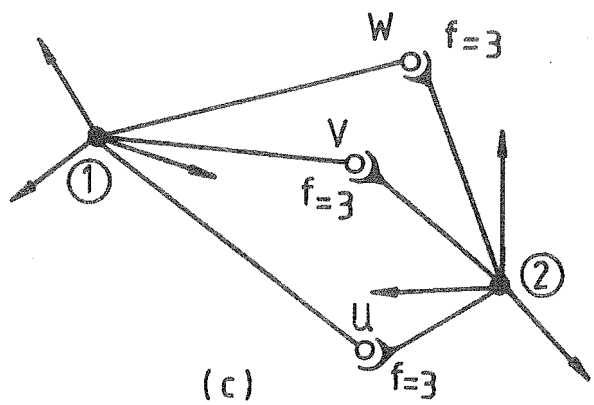
Figs 4.7d and 4.7e illustrate the case where the positions of U, V and W are found simultaneously. The prismatic pair in Fig.4.7d models the indeterminacy of the position of the contact point U considered by itself (relative to A_2, U can lie anywhere along the force line ℓ_u). The rigid triangular plate UVW, also in Fig. 4.7d, is a reminder of the invariability of the distances between U, V and W. When the corners of this plate are fitted simultaneously onto the respective force lines (Fig.4.7e), the linkage between



(a)

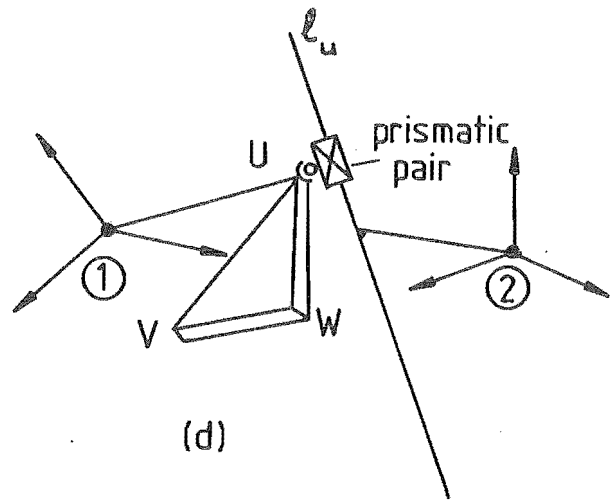


(b)

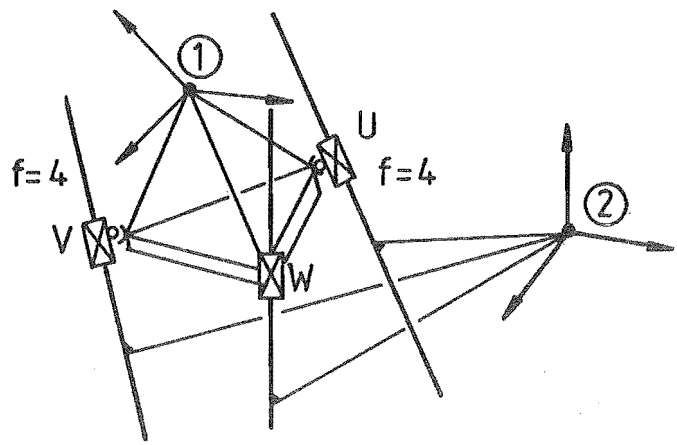


(c)

FIG 4·7 KINEMATIC MODELS OF TOUCH–FORCE SENSING (see also overleaf)



(d)



(e)

FIG 4·7 KINEMATIC MODELS OF TOUCH-FORCE SENSING (cont'd)

A_1 and A_2, for which

$$n = 2,$$

$$j = 3,$$

$$\text{and } \Sigma f = 12,$$

becomes locked, with F = 0 degrees of freedom. Again, it follows that the relative location between A_1 and A_2 then is completely specified.

4.2.1.2 Force-touch sensing

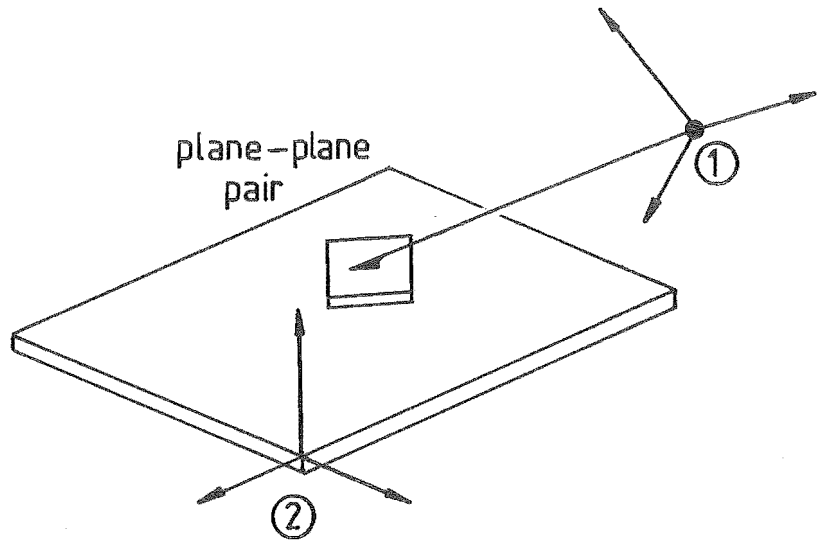
As described in Section 3.3.2.3, array A_1 is a wand capable of detecting forces applied at its tip and array A_2 consists of touch sensors mounted in three mutually orthogonal datum planes.

When the tip of the wand contacts a datum plane, the touch sensor in the latter is triggered and the plane's identity established. The location of the plane then becomes evident to the coordinate system of A_2 (which naturally, is fixed with respect to all datum planes).

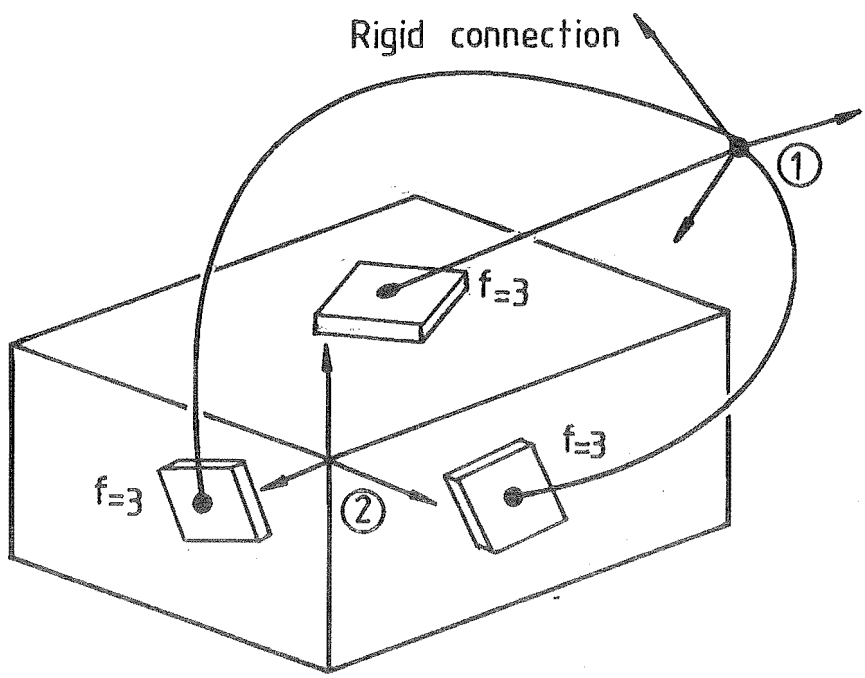
If the contact between the tip of the wand and the plane is frictionless, the contact force picked up by A_1 is always normal to the plane and therefore the orientation of the plane relative to A_1 can be found. Also, because the position of a point on the plane (namely the tip of the wand) is known, the location of the plane with respect to A_1 is completely determinable.

We can model the fact that the location of a plane is known in both the coordinate systems of A_1 and A_2 by joining them with a plane-plane pair (Fig.4.8a). Note that one half of this pair is rigidly attached to the coordinate system of A_1 and represents the plane as 'seen' by A_1; the other half of the pair is rigidly attached to the coordinate system of A_2 and represents the same plane as 'seen' by A_2. Clearly, both halves must lie flat against each other, although they, and hence A_1 and A_2, are capable of sliding and rotating relatively to each other.

By establishing the locations of 3 different planes relative to both A_1 and A_2, we arrive at the model of Fig.4.8b, where A_1 and A_2 are shown joined together with 3 plane-plane pairs. Thus, for the linkage between A_1 and A_2,



(a)



(b)

FIG 4·8 KINEMATIC MODELS OF FORCE-TOUCH SENSING

$$n = 2,$$
$$j = 3,$$
$$\text{and } \Sigma f = 9.$$

Therefore, according to Kutzbach formula,

$$F = -3$$

which again implies that A_1 is completely located with respect to A_2.

4.2.1.3 Touch-touch sensing

The sensor arrangement in this case is similar to that in force-touch sensing; A_1 is still a wand and A_2 a set of touch sensors mounted in a triad of mutually orthogonal datum planes. Unlike in force-touch sensing, however, the wand does not measure forces applied at its tip and can only detect whether touching occurs there.

Thus, when the tip of the wand contacts a datum plane, the location of the plane is defined with respect to the coordinate system of A_2 (as in force-touch sensing) while only the position of the contact point is defined with respect to the coordinate system of A_1. This is modelled in Fig.4.9a by rigidly fixing the plane to A_2 and the point (represented by a sphere), to A_1. Since the point must always lie on the plane, A_1 and A_2 are in effect, joined by a sphere-plane pair with two translational and three rotational degrees of freedom.

Figs 4.9b and 4.9c are kinematic models of the two variants of touch-touch sensing in which the tip of the wand makes six contacts with the datum planes. Hence, in both figures we find A_1 and A_2 joined together by six sphere-plane pairs. The linkages between A_1 and A_2 then have

$$n = 2,$$
$$j = 6,$$
$$\text{and } \Sigma f = 30.$$

From Kutzbach formula,

$$F = 0,$$

i.e., both linkages are structures and the relative location between A_1 and A_2 is completely defined.

If we introduce a 7th contact between the tip of the wand and the datum

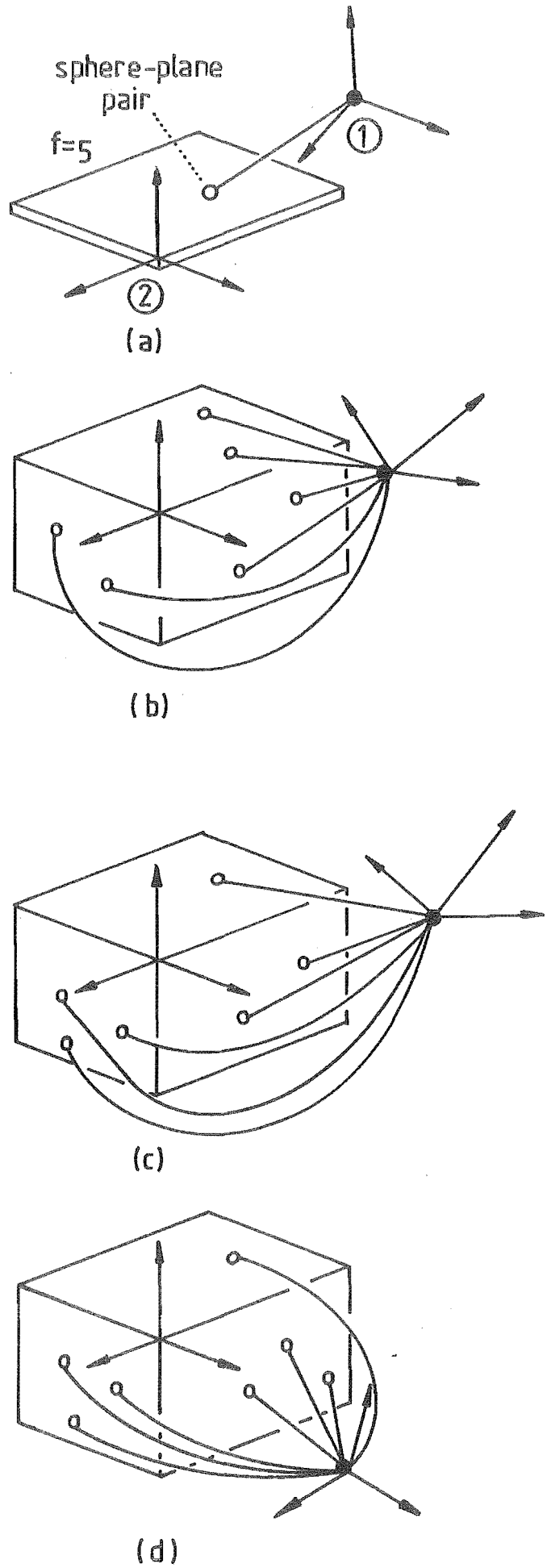


FIG 4·9 KINEMATIC MODELS OF TOUCH-TOUCH SENSING

planes, we obtain the third variant of touch-touch sensing, a model of which is shown in Fig.4.9d. The kinematic parameters of the linkage between A_1 and A_2 now become

$$n = 2,$$
$$j = 7,$$
$$\text{and } \Sigma f = 35.$$

Clearly, $F = -1$ and the relative location between A_1 and A_2 is, again, completely defined.

4.2.2 Discussion

So far, we have justified how various bilateral sensing techniques can be represented by simple kinematic models. Using these models, we have shown repeatedly that it is possible to determine the relative location of two contact-sensing arrays by comparing the responses of one array to a set of contact events with the simultaneous responses of the other array to the same set of events.

The comparison of the responses of two sensor arrays, as we have seen, is akin to the establishment of kinematic pairs, which in a figurative sense, link the two arrays together. The number of pairs, j, required to form a rigid immovable linkage between these arrays changes from one technique to another, but is invariably greater than one throughout the family of techniques that we have examined. In other words, these particular techniques all exhibit a multinary character.

Fig.4.10 is a kinematic model of yet another bilateral technique. The technique is completely arbitrary in that its model specifies neither the number j of kinematic pairs nor the nature of these pairs. Nevertheless, as with other bilateral techniques, here the number n of links and the number F of degrees of freedom are given by

$$n = 2 \qquad \text{Eq. 4.39}$$

and

$$F \leq 0. \qquad \text{Eq. 4.40}$$

Combining Eqs 4.38, 4.39 and 4.40 yields

$$6 - 6j + \Sigma f \leq 0 \qquad \text{Eq. 4.41}$$

where Σf is the total number of degrees of freedom in the j pairs of the model in question.

We shall now discuss how Eq.4.41 can be used for synthesizing new bilateral techniques as well as for establishing a general property of all bilateral techniques.

4.2.2.1 Synthesis of bilateral techniques

The procedure is to:

(i) fix j. This determines the lower limit on Σf;

(ii) compute the upper limit on Σf from Eq.4.41;

(iii) select a value for Σf from the allowable range of values between the lower and upper limits found in (i) and (ii);

(iv) decide on the nature of each of the j pairs;

(v) deduce the types of sensors and sensory responses required.

Example:

(i) Fix $j = 2$

$\therefore \Sigma f \geqslant 2.$

(ii) From Eq.4.41

$\Sigma f \leqslant 6.$

(iii) Select $\Sigma f = 2.$

(iv) Pair No.1, $f = 1$: revolute pair

Pair No.2, $f = 1$: revolute pair.

(v) A_1 : force-sensing array, for picking up the lines of action of two contact forces and the positions of 2 contact points.

A_2 : the same as for A_1.

Fig.4.11a is a kinematic model of the bilateral technique just synthesized. Kutzbach formula yields $F = -4$ for this model, i.e. A_1 and A_2 are rigidly locked against each other as expected.

Note: When synthesizing new bilateral techniques, we must be wary of possible general and 'overclosing' constraints in the kinematic models of these techniques. These constraints, which arise in a model where there are special relationships between the axes of the pairs or the dimensions of the links [84] [85], increase F beyond the value normally obtained with Kutzbach formula and therefore can cause ill-conditioning in the corresponding bilateral technique. For instance, in the model of Fig.4.11a, if the two revolute pairs become collinear (Fig.4.11b), ill-conditioning will occur since F will be equal to 1 and the relative location between A_1 and A_2 will be indeterminate. Fig.4.12 shows how the bilateral techniques analysed previously can also fail by ill-conditioning.

4.2.2.2 A theorem on bilateral sensing

Eq. 4.4.1 can be rewritten as

$$j \geq 1 + \frac{\Sigma f}{6} \qquad \text{Eq. 4.42}$$

which for a given kinematic model, defines the lower limit imposed on the number j of pairs as a function of the total number Σf of degrees of freedom in these pairs.

Since Σf is strictly positive and j is an integral quantity, Eq.4.42 implies that 2 is the absolute minimum value that j can have in the kinematic model of any arbitrary bilateral technique. In other words,

THEOREM: Bilateral sensing is inherently a multinary process.

The practical connotation of this theorem is that whenever a bilateral technique is implemented, some kind of motion will be involved, due to the need for taking several independent sensor measurements. As a bilateral technique always hinges on knowing all relative sensor motions occurring between each measurement, these motions must be performed with high accuracy. For maximum efficiency, these rather unproductive motions must also be kept

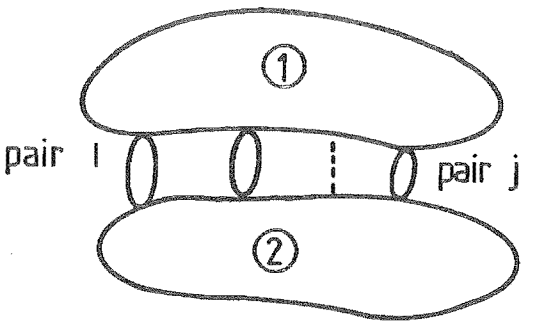
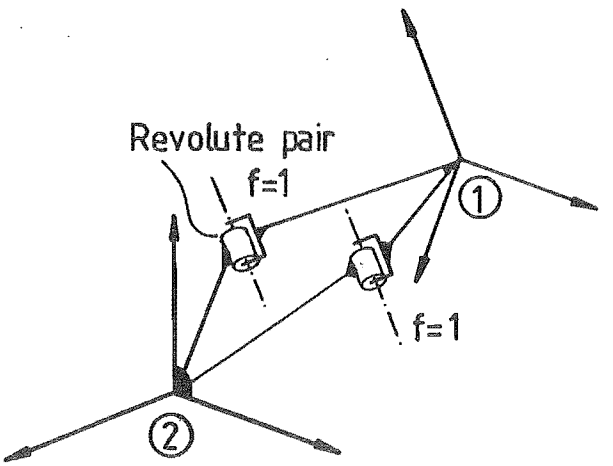
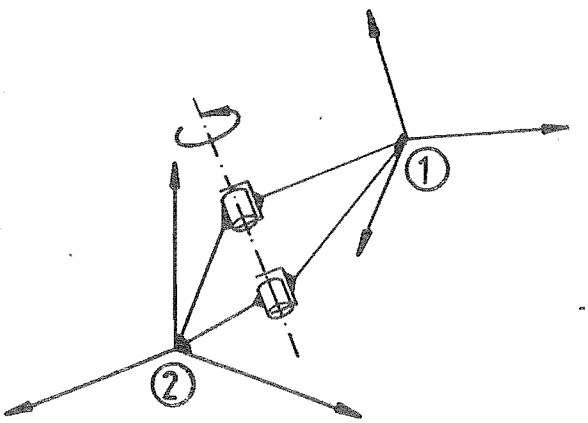


FIG 4·10 KINEMATIC MODEL OF AN ARBITRARY BILATERAL TECHNIQUE

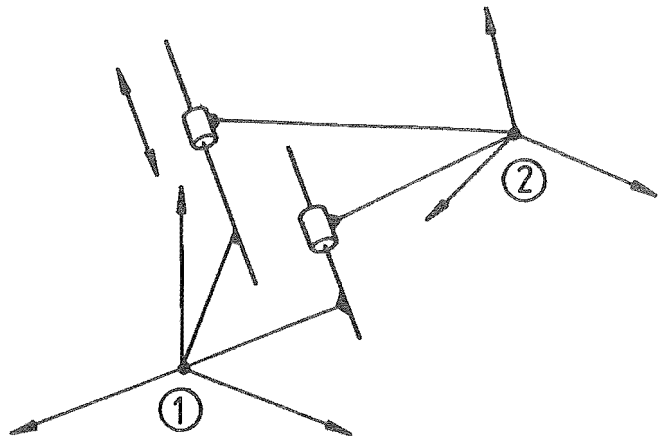


(a) normal technique

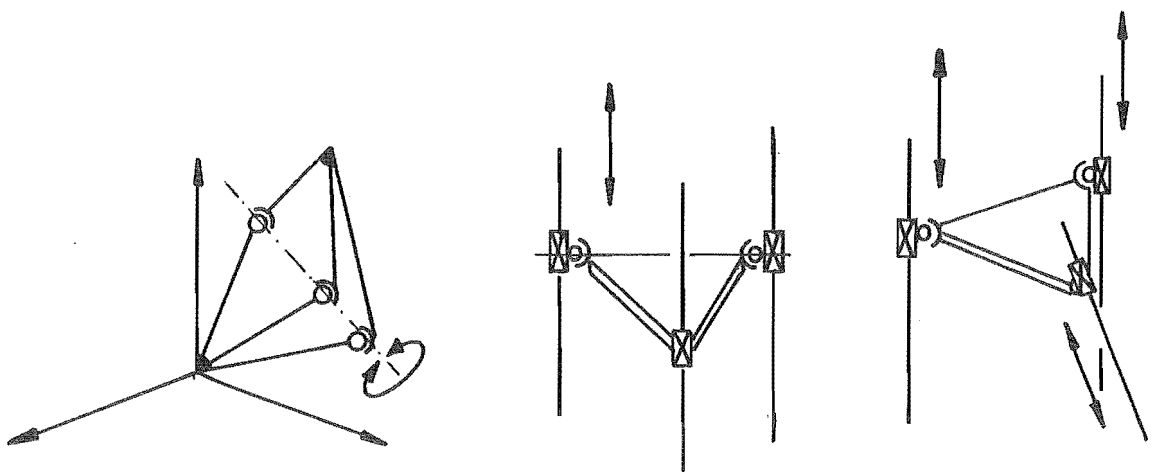


(b) Ill- conditioned technique

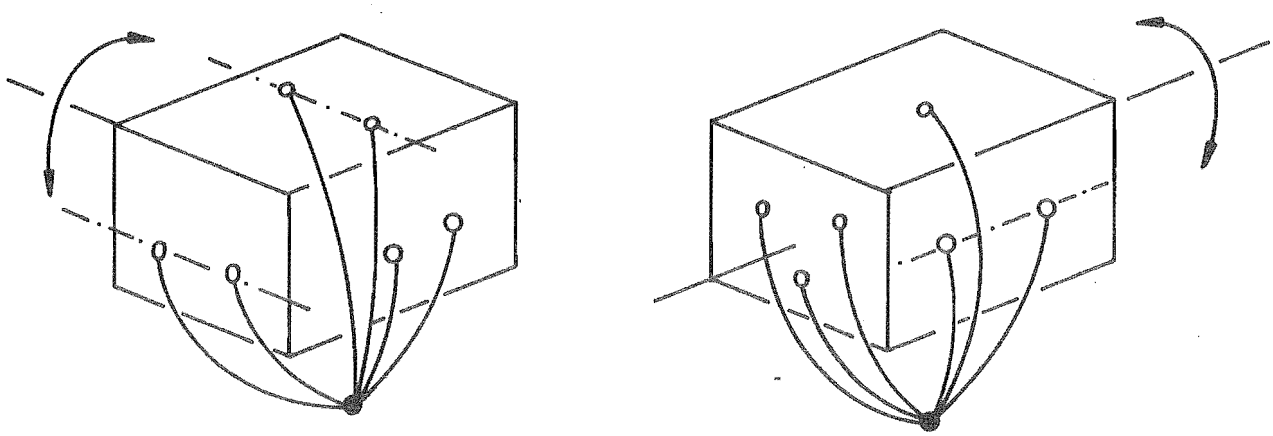
FIG 4·11 KINEMATIC MODELS OF A NEW FORCE–FORCE SENSING TECHNIQUE



(a) Force-force sensing



(b) Touch-force sensing



(c) Touch-touch sensing

FIG 4·12 VARIOUS CASES OF ILL-CONDITIONING

as small as possible. Therefore, an attribute of all assembly systems with bilateral sensing is the ability to make small and accurate motions.

4.3 SUMMARY

A special bilateral technique, force-force sensing, has been examined in detail. Through an unusual application of a simple formula by Rodrigues, a closed-form solution has been achieved for the problem of inferring the relative orientation of two force-sensing arrays from two force measurements taken in each array.

Other bilateral techniques have also been studied. A parallel has been drawn between these techniques and various kinematic methods of rigid-body location. This has enabled bilateral sensing to be treated in a unified manner by means of Kutzbach formula, a device normally employed in the synthesis or classification of kinematic linkages. The use of Kutzbach formula to generate new bilateral techniques and to prove the multinary nature of all bilateral techniques has been described.

CHAPTER 5

THE KINEMATICS OF A COMPUTER-DRIVEN ASSEMBLY MACHINE

An assembly system based on bilateral sensing may consist of a rigid work table with one or several flexible industrial robots. The robots would make the large and swift movements required for gross pick-and-place tasks, while the work table would perform the small and precise adjustments necessary for the final stage of the assembly operation.

For maximum flexibility, the robots should have at least six degrees of freedom. Their positioning errors, in the worst case, would therefore be three-dimensional. Clearly, to correct these errors, the work table should also possess no fewer than six degrees of freedom.

This chapter describes a mini-computer-controlled, six-degree-of-freedom work table suitable for investigating the proposed assembly scheme. The device resembles Gough's and Whitehall's Universal Tyre Test Machine [86], Stewart's Aircraft Simulator [87], or Koogle's motion transducer for use in the intact invitro human lumbar spine [88]. With its triangulated construction, it is robust and accurate, but at the same time light and manoeuvrable.

The main body of this chapter is divided into three sections. The first outlines the mechanical hardware; the second analyses it kinematically; and the third describes a method to synthesize its path.

5.1 HARDWARE DESCRIPTION

The work table comprises a triangular platform, six screw jacks, six stepping motors, and a triangular base (Fig.5.1a). The stepping motors are rigidly mounted at the corners of the base and drive the screw jacks via the bearing and universal joint arrangement shown in Fig. 5.1b. The jacks are grouped into three pairs, each consisting of two adjacent jacks pinned together at the top (Fig. 5.1b). Also pinned at the top of each jack pair

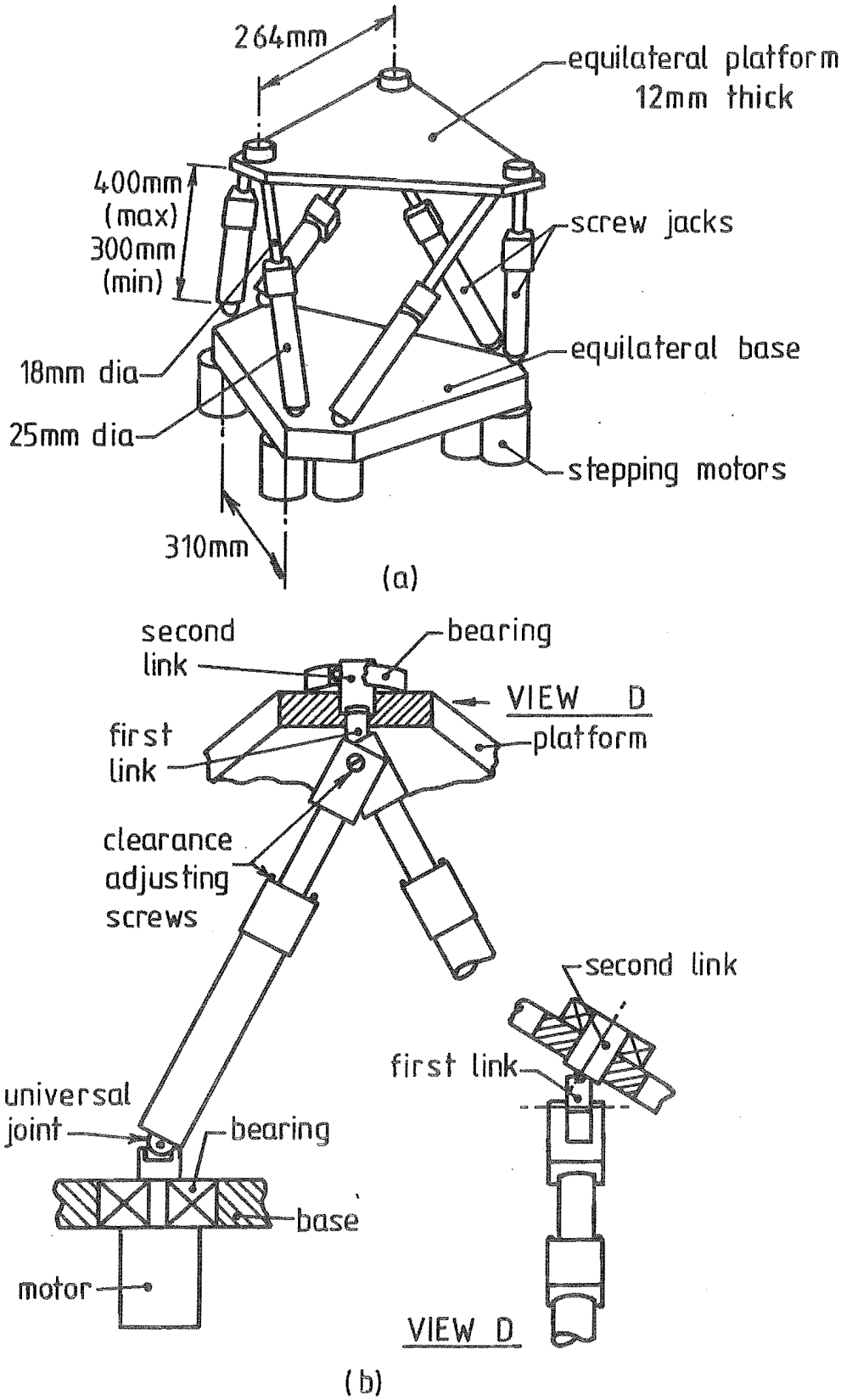


FIG. 5·1 THE WORK TABLE AND ITS MECHANICAL DETAILS.

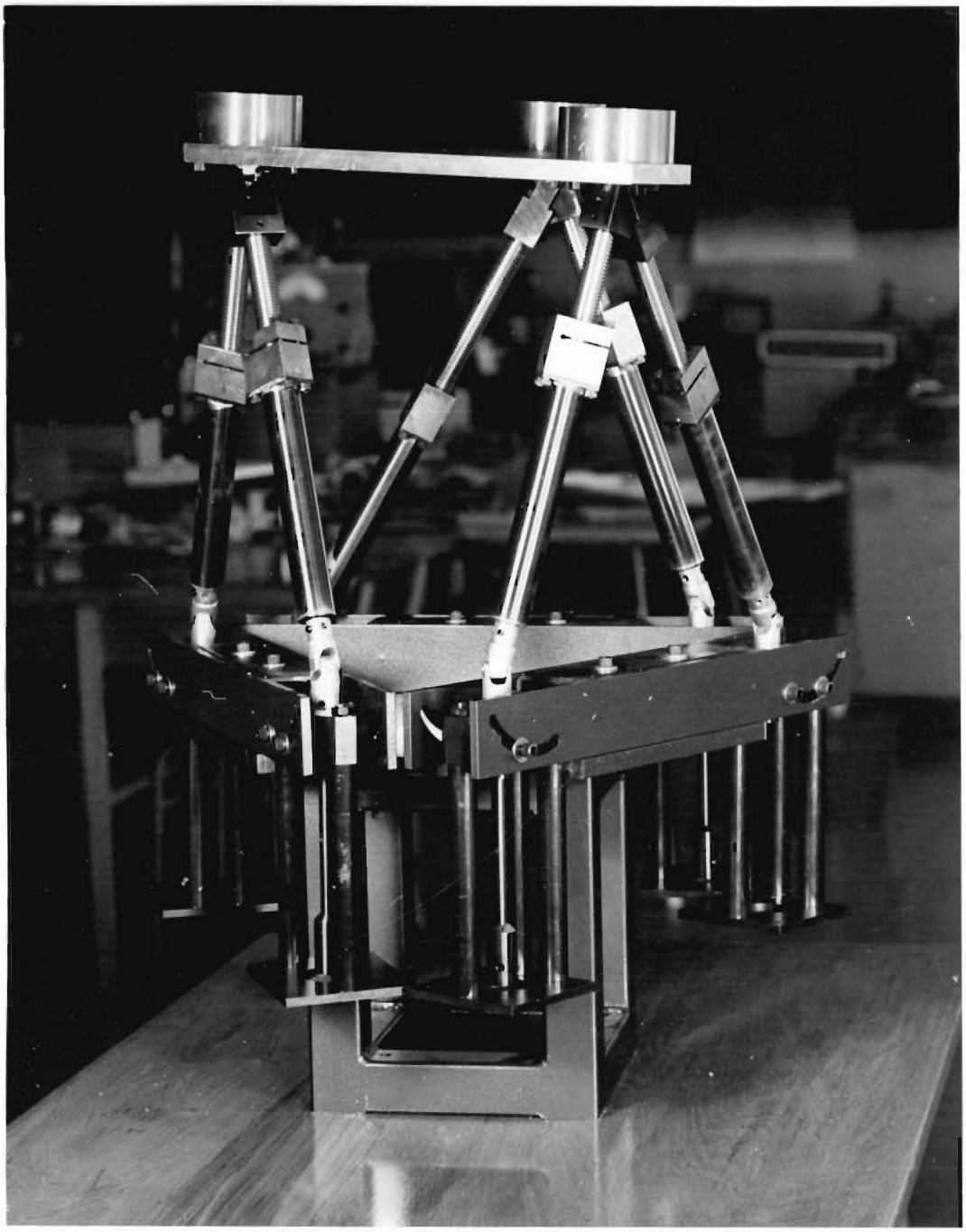


PLATE. 5·1 THE WORK TABLE

is one end of a short straight link. The other end of this link is pinned to a second link which, in turn, passes through a bearing located at a corner of the platform. The axis of the pin common to the first and second links is normal to that of the pin common to the jacks and the first link. The axes of the platform's bearings are normal to the platform plane.

The accuracy with which we can specify the location of the platform clearly depends on the rigidity of the work table's structural members, the clearances in its joints, and the dimensional tolerances of its components.

The principal dimensions of the work table are shown in Fig. 5.1. Given these dimensions, Wong [89] has found the work table's stiffness to be of the order of 10 kN/mm. That is, under the action of a 100 N-assembly force, the table will deflect by only .01 mm. Therefore, the table is more than adequately rigid for the light assembly tasks it is designed to perform.

To increase the accuracy of the table, the joints have been provided (where feasible) with means for adjusting their clearances and restricting these to an essential minimum (Fig. 5.1b).

There are two types of dimensional errors. The first is associated with uncertainties in the manufacture of the various components of the work table and the second, with the resolution of the screw jack and stepping motor arrangement. The negative effects of the first type of errors on the positioning accuracy of the system are easily counteracted by calibration. The second type of errors are minimised with the combination of fine-pitch screws and high-resolution stepping motors. In the present design, the screws have ten threads per inch and the stepping motors make two hundred steps per revolution. The jack lengths can therefore be controlled to within 0.5 thousandth of an inch (.01 mm). Preliminary tests on the positioning accuracy and repeatability of the overall system have indicated values of the same order of magnitude.

5.2 KINEMATIC ANALYSIS

This section is divided into three parts. The first part verifies that the work table has six degrees of freedom and that the platform can be located at any position and orientation relative to the stationary base. The second and third parts develop and analyse the mathematical relationships between the platform and the base.

5.2.1 Degree of Freedom and Connectivity

5.2.1.1 Degree of freedom

The number of degrees of freedom of any system is the minimum number of independent parameters required to specify its state completely. Thus the number of degrees of freedom, F, of a mechanism is the minimum number of joint variables necessary to determine the location of every link relative to a given link in the mechanism.

F, for a spatial mechanism with no general constraints, is again given by the well-known Kutzbach formula

$$F = 6(n-1) - 6j + \Sigma f \qquad \text{Eq. 5.1}$$

where n = number of links (including the fixed link)

j = number of joints (binary joints are counted as one, ternary joints as two,etc.)

Σf = total number of degrees of freedom in all the joints.

A kinematic model of the work table with all the six motors activated is shown in Fig. 5.2. For this model, clearly,

$n = 20$

$j = 24$

$\Sigma f = 36$

and since there are no general constraints, Kutzbach formula may be applied to give F equal to 6.

Hence the work table has six degrees of freedom and the six parameters

needed to specify its state may be taken as either the angles of rotation of the stepping motors or the variable lengths of the screw jacks.

Obviously when all the motors are stopped, the screw jack lengths are no longer variable, the sum Σf is reduced by 6, F is zero, and the work table becomes a rigid structure.

5.2.1.2 Connectivity

The connectivity C_{XY} between two rigid members X and Y in a mechanism is the number of degrees of relative freedom between them. We wish to show that the connectivity between the platform and the base is also six, that is, the platform may occupy any location relative to the base.

In general, there is no simple relationship between C_{XY} and F. For example, the connectivity between the lower half of any screw jack and the base is 3, the connectivity between the lower and upper halves of the same jack is 1, while F has been found to be 6.

There are techniques for determining C_{XY} based on the analysis of the screw system of the joints between X and Y [90]. However, because of symmetry, we do not need to carry out this complex analysis to determine the connectivity C_{PB} between the platform and the base.

We shall now define what we mean by symmetry and then state and prove a simple theorem that gives us C_{PB} directly.

DEFINITION: Let two rigid links X and Y be joined by joints J_1, ... J_n where J_1, ... J_n can be linkages themselves [90]. If J_1, ... J_n all have the same degree of freedom, then X and Y are said to be symmetrically joined. (Fig.5.3.)

THEOREM: If the links X and Y in a mechanism are symmetrically joined, then the relative degree of freedom, or connectivity C_{XY}, between X and Y is equal to the smaller of the number of degrees of freedom F of the whole mechanism

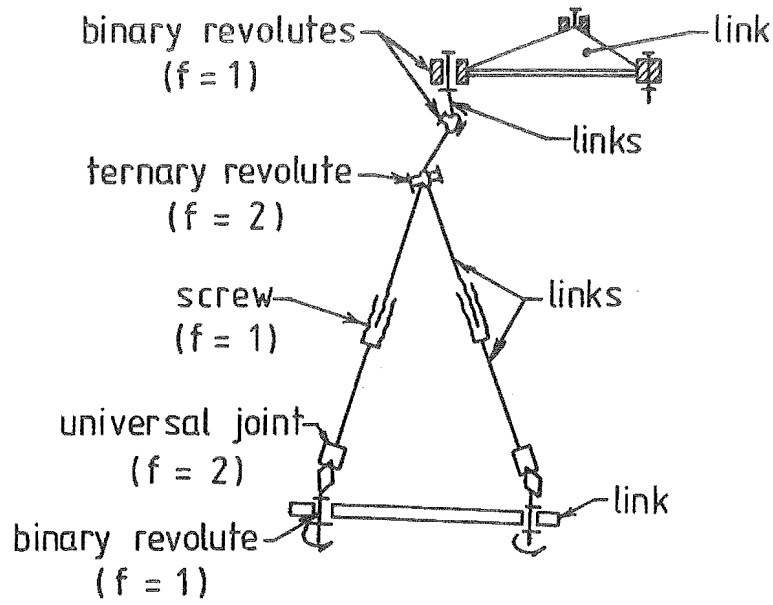
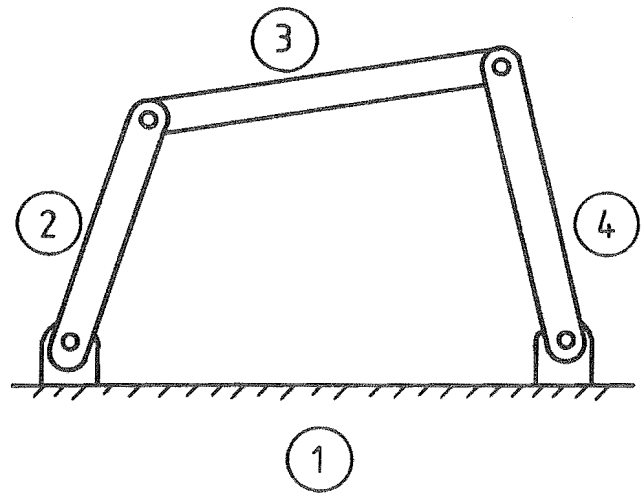


FIG 5·2 KINEMATIC MODEL (active motors)



1	&	3	are	symmetrically	joined
2	&	4	"	"	"
1	&	2	are	asymmetrically	joined
1	&	4	"	"	"
2	&	3	"	"	"
3	&	4	"	"	"

FIG 5·3 SYMMETRY IN A PLANAR FOUR-BAR LINKAGE

or the number of degrees of freedom S of the space within which the mechanism operates, i.e. $C_{XY} = \min (F,S)$.

PROOF: First consider the case F<S. (Fig. 5.4a). Because of symmetry, $C_{XY} > C_{XI}$ where C_{XI} is the connectivity between X and any link I in the mechanism.

From the definition of connectivity, Y is completely fixed relative to X if C_{XY} joint parameters are known. Because $C_{XI} < C_{XY}$, the locations relative to X of all links I are also completely determined given the C_{XY} parameters.

Thus $C_{XY} \nmid F$ Eq. 5.2

Also, since the connectivity between any two members in a mechanism cannot be greater than the number of degrees of freedom of the mechanism itself,

$C_{XY} \nmid F$ Eq. 5.3

From Eqs 5.2 and 5.3, $C_{XY} = F$

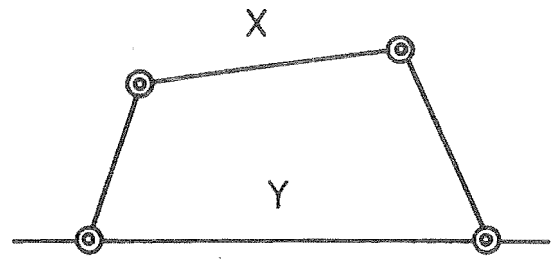
As F increases to S (Fig. 5.4b), $C_{XY} = F = S$.

When F > S (Fig. 5.4c), $C_{XY} = S$, because the number of degrees of relative freedom between any two members in a mechanism cannot exceed the number of degrees of freedom of the space in which the mechanism operates.

It is evident from the kinematic model of the work table (Fig. 5.2) that the platform and the base are symmetrically joined. As F = S = 6, the connectivity C_{PB} is also 6 and, therefore, relative to the base, the platform can assume any arbitrary position and orientation within the working space of the mechanism.

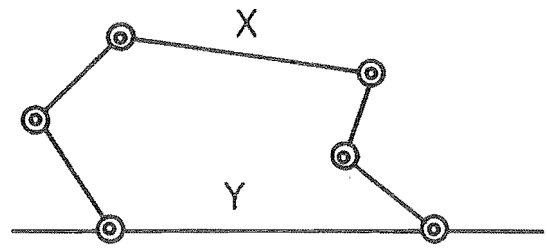
5.2.2 Platform-base Location and Parameters of the Work Table

We have shown that the platform has six degrees of freedom relative to the base. Therefore, the platform-base location may be represented by



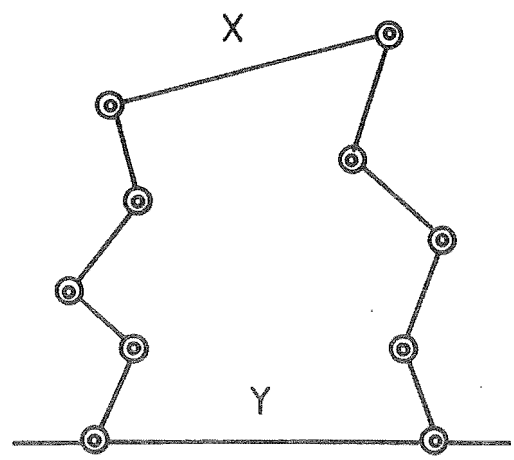
S = 3
F = 1
C_{XY} = 1

(a) F < S



S = 3
F = 3
C_{XY} = 3

(b) F = S



S = 3
F = 7
C_{XY} = 3

(c) F > S

FIG 5·4 THE CONNECTIVITY THEOREM APPLIED TO SIMPLE PLANAR LINKAGES.

six independent co-ordinates. Let $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ and $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$ be two triads of rectangular axes fixed on the platform and the base respectively. The spatial relationship between these two rigid bodies may be taken as the components of the generalised co-ordinate vector $\underset{\sim}{L}_i$

$$\underset{\sim}{L}_i = (t_{xi}, t_{yi}, t_{zi}, \phi_i, \theta_i, \psi_i)$$

where t_{xi}, t_{yi}, t_{zi} are the linear co-ordinates of the origin of the triad $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ and ϕ_i,θ_i,ψ_i, its Eulerian angles, relative to the triad $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$.

During a task, often the platform has to alter its location $\underset{\sim}{L}_i$ so that successful assembly can occur. New locations $\underset{\sim}{L}_j$'s may be computed from tactile feedback information. In order to take the platform from $\underset{\sim}{L}_i$ to $\underset{\sim}{L}_j$'s, we must alter the work table's parameters. To be able to do so, we must know how these parameters relate to the platform-base location.

As mentioned earlier, either the stepping motors' rotation angles or the screw jack lengths may be taken as the six parameters of the six-degree - of-freedom work table. For convenience, we have chosen the latter and have grouped them into the jack-length vector $\underset{\sim}{\ell}_i$

$$\underset{\sim}{\ell}_i = (\ell_{i1}, \ell_{i2}, \ldots\ldots \ell_{i6})$$

Thus, we wish to find the relationship G, such that

$$\underset{\sim}{L}_i \xrightarrow{\quad G \quad} \underset{\sim}{\ell}_i$$

or $\underset{\sim}{\ell}_i = G(\underset{\sim}{L}_i)$

The successive steps leading from $\underset{\sim}{L}_i$ to $\underset{\sim}{\ell}_i$ are described below

(a) determination of the transformation matrix $\begin{bmatrix} [R]_i & \underset{\sim}{T}_i \\ 0 & 1 \end{bmatrix}$ linking $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ and $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$ using

$$\underset{\sim}{L}_i = (t_{xi}, t_{yi}, t_{zi}, \phi_i, \theta_i, \psi_i).$$

It is well known (see [78] for instance) that

$$[R]_i = \begin{bmatrix} \cos\phi_i \cos\theta_i \cos\psi_i - \sin\phi_i \sin\psi_i & -\cos\phi_i \cos\theta_i \sin\psi_i - \sin\phi_i \cos\psi_i & \cos\phi_i \sin\theta_i \\ \sin\phi_i \cos\theta_i \cos\psi_i + \cos\phi_i \sin\psi_i & -\sin\phi_i \cos\theta_i \sin\psi_i + \cos\phi_i \cos\psi_i & \sin\phi_i \sin\theta_i \\ -\sin\theta_i \cos\psi_i & \sin\theta_i \sin\psi_i & \cos\theta_i \end{bmatrix} \quad \text{Eq. 5.4a}$$

and $\underset{\sim}{T}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}$ Eq. 5.4b

and that the position vector $\underset{\sim}{P}$ and $\underset{\sim}{p}$ of a point P in $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$ and $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ respectively, are related by

$$\begin{bmatrix} \underset{\sim}{P} \\ 1 \end{bmatrix} = \begin{bmatrix} [R]_i & \underset{\sim}{T}_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{p} \\ 1 \end{bmatrix}$$

(b) Determination of the platform's unit normal vector and the position of the platform's corner E (Fig. 5.5).

The platform's normal $\underset{\sim}{\hat{n}}$ is given by

$$\underset{\sim}{\hat{n}} = \begin{bmatrix} \cos\phi_i \sin\theta_i \\ \sin\phi_i \sin\theta_i \\ \cos\theta_i \end{bmatrix}, \text{ i.e. the third column of } [R]_i$$

Let $\underset{\sim}{E}$ and $\underset{\sim}{e}$ be the position vectors of the platform's corner E in $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$ and $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ respectively. $\underset{\sim}{e}$ is a constant vector determined by the platform's dimensions.

From (a)

$$\begin{bmatrix} \underset{\sim}{E} \\ 1 \end{bmatrix} = \begin{bmatrix} [R]_i & \underset{\sim}{T}_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{e} \\ 1 \end{bmatrix}$$

(c) Determination of $\hat{\underset{\sim}{w}}$, the unit direction vector of the axis of the pin common to the first link CD and second link DE.

As $\hat{\underset{\sim}{w}}$ is a unit vector, $\left|\hat{\underset{\sim}{w}}\right| = 1$ Eq. 5.5

Also, $\hat{\underset{\sim}{w}}$ is normal to $\hat{\underset{\sim}{n}}$ $\hat{\underset{\sim}{w}}.\hat{\underset{\sim}{n}} = 0$ Eq. 5.6

From Fig.5, $\hat{\underset{\sim}{w}}$ is coplanar with $(\underset{\sim}{E} - v\hat{\underset{\sim}{n}} - \underset{\sim}{A})$ and $(\underset{\sim}{B}-\underset{\sim}{A})$

$$\hat{\underset{\sim}{w}} = \lambda(\underset{\sim}{E} - v\hat{\underset{\sim}{n}} - \underset{\sim}{A}) + \mu(\underset{\sim}{B} - \underset{\sim}{A}) \qquad \text{Eq. 5.7}$$

where v is the length of the link DE and $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are constant position vectors.

Let $\underset{\sim}{M} = \underset{\sim}{E} - v\hat{\underset{\sim}{n}} - \underset{\sim}{A}$ Eq. 5.8

and $\underset{\sim}{AB} = \underset{\sim}{B} - \underset{\sim}{A}$ Eq. 5.9

Note that, given the dimensions of the work table, $\underset{\sim}{M}$ is never parallel to $\underset{\sim}{AB}$.

If $\underset{\sim}{AB}.\hat{\underset{\sim}{n}} \neq 0$, combining Equations 5.5, 5.6, 5.7, 5.8 and 5.9 gives

$$\lambda = \pm \frac{1}{\left| \underset{\sim}{M} - \frac{\underset{\sim}{M}.\hat{\underset{\sim}{n}}}{\underset{\sim}{AB}.\hat{\underset{\sim}{n}}} \underset{\sim}{AB} \right|}$$

and

$$\mu = - \frac{\underset{\sim}{M}.\hat{\underset{\sim}{n}}}{\underset{\sim}{AB}.\hat{\underset{\sim}{n}}} \lambda$$

If $\underset{\sim}{AB}.\hat{\underset{\sim}{n}} = 0$, since $\hat{\underset{\sim}{n}}$ cannot be normal to the plane $(\underset{\sim}{M}, \underset{\sim}{AB})$, $\hat{\underset{\sim}{w}}$ and $\underset{\sim}{AB}$ have the same direction

$$\lambda = 0$$

$$\mu = \pm \left| \frac{1}{\underset{\sim}{AB}} \right|$$

Since the sense of $\hat{\underset{\sim}{w}}$ is immaterial to the ensuing calculation, we are free to choose the sign of λ (in the first case) or μ (in the second case).

(d) Determination of $\hat{\underset{\sim}{u}}$, the direction vector of the first link CD. As $\hat{\underset{\sim}{u}}$

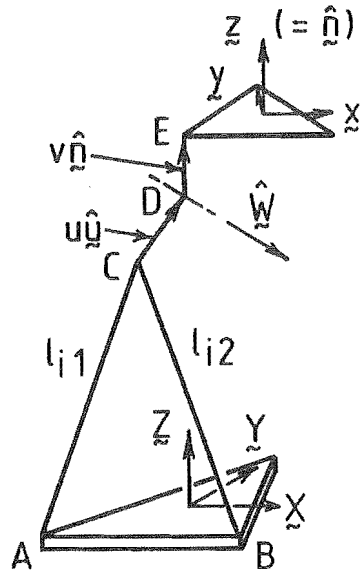


FIG 5·5 LINKAGE GEOMETRY

is a unit vector,

$$|\hat{u}| = 1 \qquad \text{Eq. 5.10}$$

$\hat{u}$ is normal to $\hat{w}$,

$$\hat{u}.\hat{w} = 0 \qquad \text{Eq. 5.11}$$

$\hat{u}$, M, and AB are coplanar

$$\hat{u} = \alpha M + \beta AB \qquad \text{Eq. 5.12}$$

Since both $AB.\hat{w}$ and $\left|M - \frac{M.\hat{w}}{AB.\hat{w}} AB\right|$ are non zero (unless the work table is to undergo severe contortion), Eqs 5.10, 5.11 and 5.12 yield

$$\alpha = \pm \frac{1}{\left|M - \frac{M.\hat{w}}{AB.\hat{w}} AB\right|}$$

$$\beta = - \frac{M.\hat{w}}{AB.\hat{w}} \alpha$$

Furthermore, α must be positive since within the physical working space of the platform, $\hat{u}$ has a positive component in the Z direction, while AB has no component in this direction.

(e) Determination of jack lengths.

Let u denote the length of the first link and C the position vector of the point C, common to the first link and the jack pair.

$$C = E - v\hat{n} - u\hat{u}$$

$$\ell_{i1} = |C - A|$$

$$\ell_{i2} = |C - B|$$

Similarly, the components ℓ_{i3} ..... ℓ_{i6} of ℓ_i may be found by repeating the steps (b),.....(e) of the above algorithm.

Thus, for an arbitrary location L_i of the platform we can derive a unique set of corresponding jack lengths ℓ_i and therefore we know how to change these lengths to achieve any desired platform motion.

As a corollary of the unique correspondence of $\underset{\sim}{\ell}_i$ to $\underset{\sim}{L}_i$, when we immobilise the platform relative to the base, the whole work table freezes into a rigid structure with zero degree of freedom. From Voinea and Atanasiu [91], the connectivity, C_{PB}, between the platform and the base is the difference between the work table's numbers of degrees of freedom, F, before and after the platform is immobilised. Since before immobilisation, F is 6, and afterwards it is 0, C_{PB} is 6, which verifies the result in the previous sub-section.

NOTE: In the electronic control system we have developed for the work table, the actual position control parameters are not the jack lengths, but the rotation angles of the stepping motors. Fig.5.6 explains how these angles can be found when the jack lengths and positions are known.

5.2.3 Parameters of the work table and platform-base location

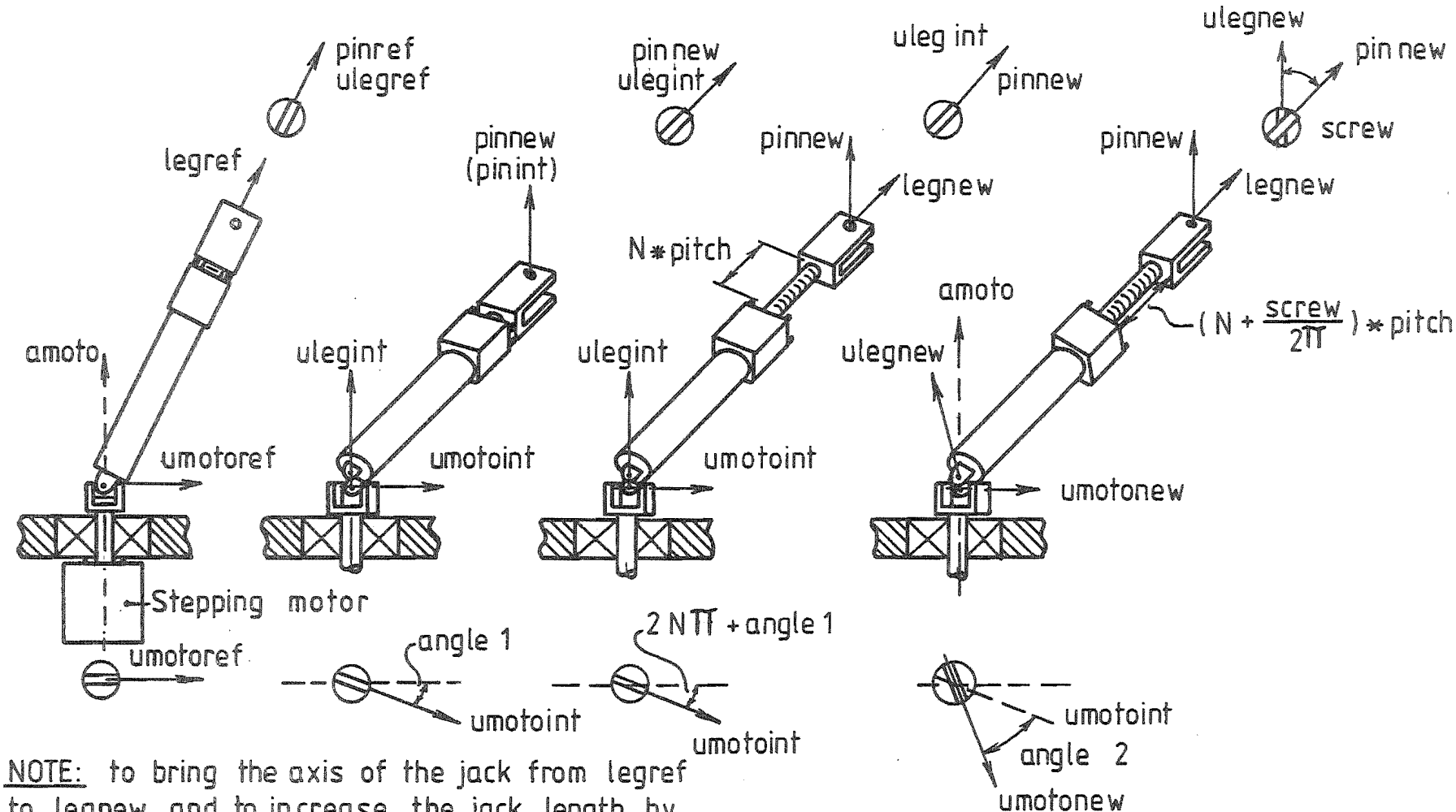
We know how to move the platform from one location to another but still need to establish the location from which the very first motion begins. In this sub-section therefore, we are interested in finding the inverse relation G^{-1}, such that

$$\underset{\sim}{\ell}_i \xrightarrow{\quad G^{-1} \quad} \underset{\sim}{L}_i$$

or $$\underset{\sim}{L}_i = G^{-1}\,(\underset{\sim}{\ell}_i)$$

since with G^{-1} we can determine the location of the platform for any set of input parameters, in particular its initial location $\underset{\sim}{L}_o$ for the initial set of jack lengths $\underset{\sim}{\ell}_o$.

It is readily appreciated that, due to the work table's complex kinematics, G^{-1} does not possess a simple closed form but must be implied from the closure and constraint equations of the linkage system. In what follows, we shall derive these equations and indicate how their numerical solution can be used for studying G^{-1}.



NOTE: to bring the axis of the jack from legref to legnew and to increase the jack length by $(N + \frac{screw}{2\Pi}) * pitch$, the stepping motor must turn $-(2N\Pi + angle\ 1 + angle\ 2)$.

FIG. 5·6 MOTOR ROTATION AND CHANGE IN SCREW JACK CONFIGURATION

5.2.3.1 Closure and constraint equations

Fig.5.7 shows a kinematic model of the work table with locked stepping motors. Since the motors are inactive, the jack lengths $\underset{\sim}{\ell}_i$ are constant, and the jack pairs form rigid triangular structures hinged about the axes $\underset{\sim}{AB}$ to the base of the work table. The angles Φ are the unknown angles between each jack-pair plane and the base plane. The angles Ψ are the unknown angles between the first links CD and the axes AB.

ℓ_{i1}, ℓ_{i2}, Φ, and Ψ, are used to compute the position vector $\underset{\sim}{D}_1$ and the axis $\underset{\sim}{\hat{W}}1$

$$\begin{bmatrix} \underset{\sim}{D}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} [R(\gamma_1)] & \underset{\sim}{T}_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{D}_1^* \\ 1 \end{bmatrix}$$

and $$\begin{bmatrix} \underset{\sim}{\hat{W}}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} [R(\gamma_1)] & \underset{\sim}{T}_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{\hat{W}}_1^* \\ 1 \end{bmatrix}$$

where $$[R(\gamma 1)] = \begin{bmatrix} \cos\gamma_1 & -\sin\gamma_1 & 0 \\ \sin\gamma_1 & \cos\gamma_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

γ_1 = a constant angle

$\underset{\sim}{T}_1$ = a constant vector

$$\underset{\sim}{D}_1^* = \begin{bmatrix} -u\cos\Psi_1 & + C_1 \\ u\cos\Phi_1 \sin\Psi_1 & + h_1 \cos\Phi_1 \\ u\sin\Phi_1 \sin\Psi_1 & + h_1 \sin\Phi_1 \end{bmatrix}$$

$$C_1 = 0.5\ (AB + \frac{\ell_{i1}^2 - \ell_{i2}^2}{AB})$$

$$h_1 = (\ell_{i1}^2 - C_1^2)^{0.5}$$

$$\underset{\sim}{W}_1^* = \begin{bmatrix} \sin\Psi_1 & \\ \cos\Phi_1 & \cos\Psi_1 \\ \sin\Phi & \cos\Psi \end{bmatrix}$$

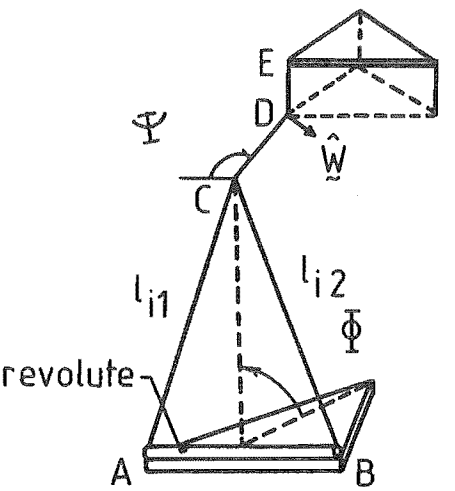


FIG 5·7 KINEMATIC MODEL (LOCKED MOTORS)

Similar expressions for $(\underset{\sim}{D}_2, \hat{\underset{\sim}{W}}_2)$ and $(\underset{\sim}{D}_3, \hat{\underset{\sim}{W}}_3)$ can also be derived in terms of $(\ell_{i3}, \ell_{i4}, \Phi_2, \Psi_2)$ and $(\ell_{i5}, \ell_{i6}, \Phi_3, \Psi_3)$.

$\underset{\sim}{\ell}_i$, Φ's and Ψ's can now be related by the following equations

$$(\underset{\sim}{D}_1\underset{\sim}{D}_2)^2 = (\underset{\sim}{D}_2\underset{\sim}{D}_3)^2 = (\underset{\sim}{D}_3\underset{\sim}{D}_1)^2 = T^2 \qquad \text{Eq. 5.13}$$

where T is the length of the side of the equilateral platform

and $\hat{\underset{\sim}{W}}_1.\hat{\underset{\sim}{n}} = \hat{\underset{\sim}{W}}_2.\hat{\underset{\sim}{n}} = \hat{\underset{\sim}{W}}_3.\hat{\underset{\sim}{n}} = 0 \qquad \text{Eq. 5.14}$

where $\hat{\underset{\sim}{n}}$, the normal to the platform is given

by $$\hat{\underset{\sim}{n}} = \frac{\underset{\sim}{D}_1\underset{\sim}{D}_2 \times \underset{\sim}{D}_1\underset{\sim}{D}_3}{|\underset{\sim}{D}_1\underset{\sim}{D}_2 \times \underset{\sim}{D}_1\underset{\sim}{D}_3|}$$

Equations 5.13 express the closure conditions of the linkage system, and Equations 5.14, the constraints on the terminal link axes. These equations form a set of six independent non-linear equations. If they are solved for the unknown Φ's and Ψ's, then the position vectors $\underset{\sim}{D}_1$, $\underset{\sim}{D}_2$, $\underset{\sim}{D}_3$ can be determined, the plane $D_1D_2D_3$ found, the points E located, and the platform's position and orientation $\underset{\sim}{L}_i$ completely specified.

5.2.3.2 Numerical solution

Brown's algorithm for solving non-linear systems [77] can be used to extract Φ's and Ψ's from the above closure and constraint equations. To apply the algorithm, these equations are first re-arranged into the form

$$f_k(\Phi_1, \Phi_2, \Phi_3, \Psi_1, \Psi_2, \Psi_3) = 0, \quad k = 1, \ldots. 6$$

An initial guess is then made for the Φ's and Ψ's. The algorithm successively expands each of the f_k's into a linearised Taylor's series about this initial guess, and employs Gaussian elimination and back substitution to obtain improved guesses. The iterative process is said to converge when a set of Φ's and Ψ's is found such that all the f_k's are smaller than a predetermined limit, or when a 'fixed point' $(\Phi_1^*, \Phi_2^*, \Phi_3^*, \Psi_1^*, \Psi_2^*, \Psi_3^*)$ is reached.

If the system f_k is well-behaved and the initial guess sufficiently close to a root, convergence to this root will take place. Since, in general, there is more than one real root, the problem is to make a guess close to that root which corresponds to a physically possible work table configuration. If a cluster of roots exist about the physical root, the location of the platform may not be determinable numerically.

A simple technique known as deflation, has been used to test for the indeterminacy condition. 'Deflation' stands for a class of methods for finding further roots of a non-linear equation in addition to those found during earlier calculations. Brown's and Gearhart's uniform-norm deflation method [92] has been adopted. This technique excludes a previously found root $(\Phi_{1old}, \ldots\ldots \Psi_{3old})$ by forming the new system of equations f_k^* at each iteration, where

$$f_k^* = \frac{f_k}{\max\left(|\Phi_1 - \Phi_{1old}|, \ldots. |\Psi_3 - \Psi_{3old}|\right)} \quad k = 1, \ldots. 6$$

It is easily seen how the original system f_k has been deflated, i.e. the new system f_k^* no longer tends to zero values as $(\Phi_{1old}, \ldots \Psi_{3old})$ is approached but still contains all the other roots of the f_k's. One of these roots, say $(\Phi_{1new}, \ldots. \Psi_{3new})$, can then be extracted from the system f_k^* in the normal way, using Brown's algorithm. When $(\Phi_{1new}, \ldots. \Psi_{3new})$ is found, it in turn is excluded from f_k^* and a new system f_k^{**} is formed. The process is repeated until either all the roots are found or the algorithm fails to converge.

Using this deflation technique, we have been able to obtain up to nine roots from the system f_k for a given set of jack lengths $\underset{\sim}{\ell}_i$. However, by an exhaustive testing procedure, we have shown that only one of these roots is feasible for any $\underset{\sim}{\ell}_i$, i.e., indeterminacy is not a problem, if we restrict the range of G^{-1} to the set of physically allowable platform locations $\underset{\sim}{L}_i$, and the domain of G^{-1}, to the set of $\underset{\sim}{\ell}_i$'s such that

$$\ell_{min} \leq \ell_{ij} \leq \ell_{max} \qquad j = 1, \ldots\ldots 6$$

where ℓ_{min} and ℓ_{max} are the fully retracted and fully extended jack lengths shown in Fig. 5.1a.

5.3 PATH SYNTHESIS

Suppose that, in order to move the platform from $\underset{\sim}{L}_i$ to $\underset{\sim}{L}_j$, the jack lengths must change from $\underset{\sim}{\ell}_i$ to $\underset{\sim}{\ell}_j$. If the difference $(\underset{\sim}{\ell}_j - \underset{\sim}{\ell}_i)$ is used directly to drive the stepping motors, the platform will eventually reach $\underset{\sim}{L}_j$ but its motion will be zigzagged and wobbly due to the non-linear nature of G^{-1}. The imperfections in the platform's path will not be perceptible if $\underset{\sim}{L}_j$ is sufficiently close to $\underset{\sim}{L}_i$, but may be unacceptable in large movements of the platform. However, by breaking such macro-movements into elementary portions, we can construct any desired path for the platform, the smoothness and accuracy of the path depending on the number of portions it comprises.

In this section we shall outline a method for breaking a macro-movement into identical micro-components. The method is best explained with transformation matrices. We recall that, in representing the location i of the platform relative to the base, we can replace the vector $\underset{\sim}{L}_i$ with the transformation matrix $\begin{bmatrix} [R]_i & \underset{\sim}{T}_i \\ 0 & 1 \end{bmatrix}$. $[R]_i$, given by Eq. 5.4a, is the orthogonal matrix of the direction cosines of the platform-fixed triad $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ relative to the base-fixed triad $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$. $\underset{\sim}{T}_i$, given by Eq. 5.4b, is the position vector of the origin of $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ relative to $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$. Thus, a macro-movement of $(\underset{\sim}{x},\underset{\sim}{y},\underset{\sim}{z})$ from i to j, as observed in $(\underset{\sim}{X},\underset{\sim}{Y},\underset{\sim}{Z})$ may be represented by the transformation matrix $\begin{bmatrix} [R]_{ij} & \underset{\sim}{T}_{ij} \\ 0 & 1 \end{bmatrix}$ such that

$$\begin{bmatrix} [R]_j & \underset{\sim}{T}_j \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} [R]_{ij} & \underset{\sim}{T}_{ij} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} [R]_i & \underset{\sim}{T}_i \\ 0 & 1 \end{bmatrix}$$

It can be easily verified (see [78], for example), that

$$[R]_{ij} = [R]_j \; [R]_i^T$$

$$\underset{\sim}{T}_{ij} = \underset{\sim}{T}_j - [R]_{ij} \; \underset{\sim}{T}_i$$

where $[R]_i^T$ is the transpose of $[R]_i$.

Breaking the macro-movement ij into fine components, therefore, means finding micro-transformations $\begin{bmatrix} [r]_k & \underset{\sim}{t}_k \\ 0 & 1 \end{bmatrix}$ such that

$$\prod_{k=N}^{1} \begin{bmatrix} [r]_k & \underset{\sim}{t}_k \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} [R]_{ij} & \underset{\sim}{T}_{ij} \\ 0 & 1 \end{bmatrix}$$

where N = number of portions in the path to be synthesized.

Again, it can be verified that, in order to obtain identical micro-movements (Fig.5.8)

$$[r]_k = [R]_{ij}^{\frac{1}{N}} \qquad k = 1, \ldots\ldots\ldots N$$

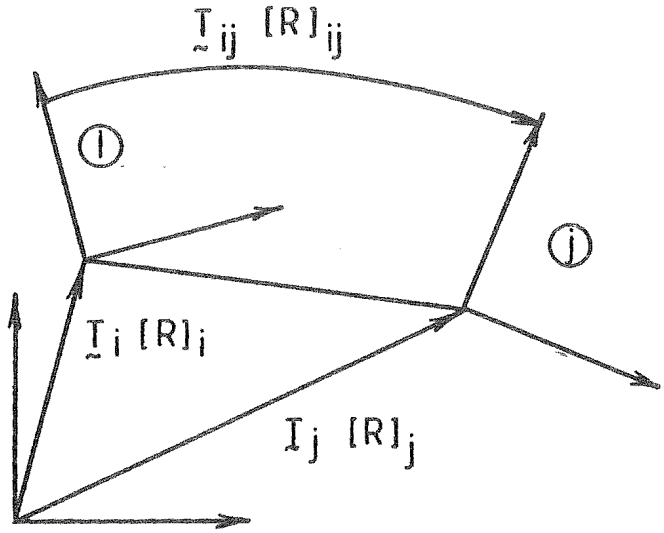
$$\underset{\sim}{t}_k = \underset{\sim}{a}_k - [r]_k \; \underset{\sim}{a}_{k-1} \qquad k = 1, \ldots\ldots\ldots N$$

where $$\underset{\sim}{a}_k = \underset{\sim}{T}_i + \frac{k}{N} (\underset{\sim}{T}_j - \underset{\sim}{T}_i) \quad k = 1, \ldots\ldots\ldots N$$

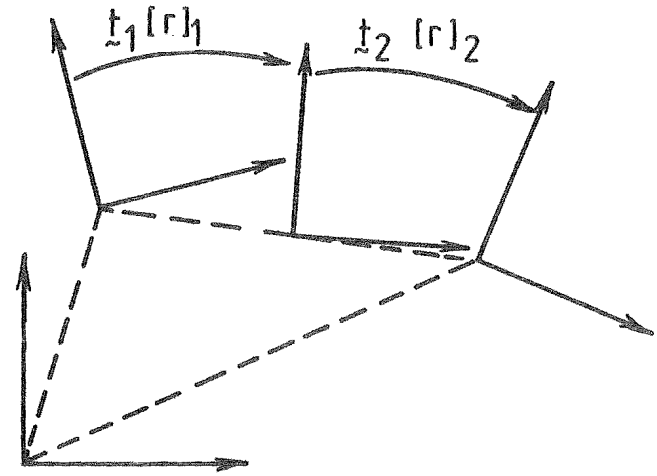
$[r]_k$ or $[R]_{ij}^{\frac{1}{N}}$, the Nth root of the orthogonal matrix $[R]_{ij}$, is readily derived from $[R]_{ij}$, if we recall that $[R]_{ij}$ represents a rotation about an axis $\hat{\underset{\sim}{a}}$ through an angle δ, and $[r]_k$, a rotation about the same axis through $\frac{\delta}{N}$. If $[R]_{ij}$ is re-written in the form of the Euler matrix [78]

$$\begin{bmatrix} \ell^2 + (1 - \ell^2)\cos\delta & \ell m\,(1 - \cos\delta) - n\sin\delta & \ell n\,(1 - \cos\delta) + m\sin\delta \\ m\ell\,(1 - \cos\delta) + n\sin\delta & m^2 + (1 - m^2)\cos\delta & mn\,(1 - \cos\delta) - \ell\sin\delta \\ n\ell\,(1 - \cos\delta) - m\sin\delta & nm\,(1 - \cos\delta) + \ell\sin\delta & n^2 + (1-n^2)\cos\delta \end{bmatrix}$$

where ℓ, m, and n, the components of $\hat{\underset{\sim}{a}}$, are derived from the elements of $[R]_{ij}$



(a) macro movement of platform from i to j.



(b) division of macro movement into two identical micro movements

FIG 5·8 PATH SYNTHESIS EXAMPLE.

as shown in [78], then $[r]_k$ is obtained by replacing δ by $\frac{\delta}{N}$ in the above matrix.

5.4 SUMMARY

This chapter has described a work table used in a study into mechanised assembly. It has checked that the mechanism has six degrees of freedom and double checked that its movable platform can be located at any position and orientation relative to its stationary base. The combined mathematical and numerical analyses have shown that there is a one-to-one correspondence between the platform's position and orientation and the input parameters of the linkage system, which is a necessary condition for controllable platform movements. The division of these movements into micro elements to build up smooth and accurate paths has also been outlined.

Among the various techniques mobilised in this chapter, of special interest is, perhaps, the combination of Brown's and Gearhart's numerical deflation and Brown's equation-solving algorithm. This combination may be useful for synthesizing new mechanisms as well as uncovering their displacement characteristics, such as dead ends, and change points etc.

Finally, note that the mechanism analysed in this chapter can have other applications in robotics. For example, it can be scaled up and used as a mobile base for mounting a robot, or 'shrunk' down and adapted into a universal six-degree-of-freedom wrist of a manipulator.

CHAPTER 6

A COMPLIANT WRIST FOR AN ASSEMBLY ROBOT

In this Chapter we shall investigate how a device, which is essentially a modified and miniaturised version of the work table studied in Chapter 5, can be used as a flexible wrist in an assembly robot.

We shall begin with a brief description of the device and then proceed to analyse it both kinematically and structurally. Our kinematic analysis will (i) verify that, like the work table, the device also possesses six degrees of freedom, and (ii) yield the compatibility matrix which relates the small displacements of the device in a 'local' coordinate system with those in a 'global' cordinate system. The compatibility matrix will subsequently be employed in our structural analysis to compute the 'global' stiffness and flexibility matrices of the device. These, in turn, will be employed to prove the unique ability of the device to assist a robot in performing a difficult assembly task - that of inserting cylindrical pegs into cylindrical holes with small clearances.

6.1 HARDWARE DESCRIPTION

As shown in Fig.6.1, the device is a triangulated arrangement of linkages very similar to that adopted for the work table; there is a base, a platform, and six 'legs'. The main difference here is that the legs are spring-loaded rams rather than motor-driven screw-jacks. Also, the connections between the legs and the platform (or the base) are realised with spheric pairs, rather than universal joints or other combinations of revolute pairs.

6.2 KINEMATIC ANALYSIS

6.2.1 Degrees of Freedom and Connectivity

6.2.1.1 Degrees of freedom

Fig.6.2 is a kinematic model of the device. Clearly the kinematic parameters are

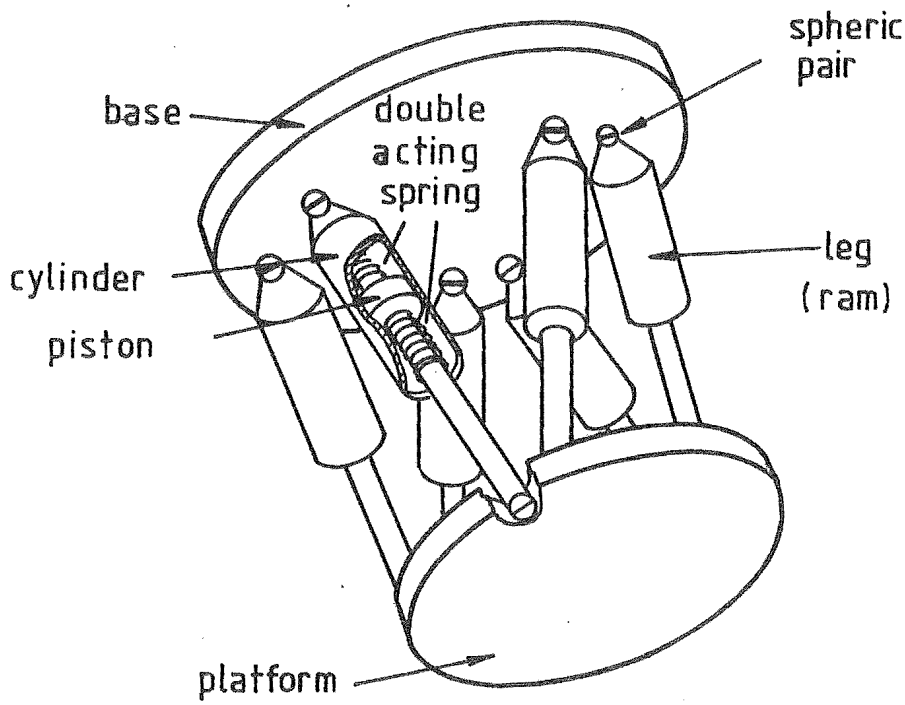


FIG 6·1 DETAILS OF THE COMPLIANT DEVICE

$$n = 14$$

$$j = 18$$

and $$\Sigma f = 48.$$

Applying Kutzbach formula in the usual way, we find the number of degrees of freedom of the device as

$$F = 18.$$

However, the device owes six out of these eighteen degrees of freedom to the ability of the rams to rotate about their own axes (Fig.6.3a), and six others, to the ability of each half of a ram to spin relatively to the other half (Fig.6.3b). Thus, the device possesses twelve degrees of freedom which cannot have any effect on its overall displacement. These degrees of freedom are known as 'redundant' degrees of freedom [84]. By subtracting the number of redundancies from F, we obtain the number of active degrees of freedom in the device as

$$F_{active} = 6.$$

(Note that F_{active} will change if general or overclosing constraints exist in the device, as illustrated in Fig.6.4.)

6.2.1.2 Connectivity

As with the work table, the platform and the base in the device are again symmetrically joined. By virtue of the Connectivity Theorem proved in Chapter 5, the number of relative degrees of freedom, or the connectivity C_{PB}, between the platform and the base is

$$C_{PB} = 6.$$

In other words, relative to the base, the platform can take up any arbitrary position and orientation within the working space of the device.

6.2.2 Compatibility Matrix

To simplify the analysis, we shall assume that pairs of adjacent rams

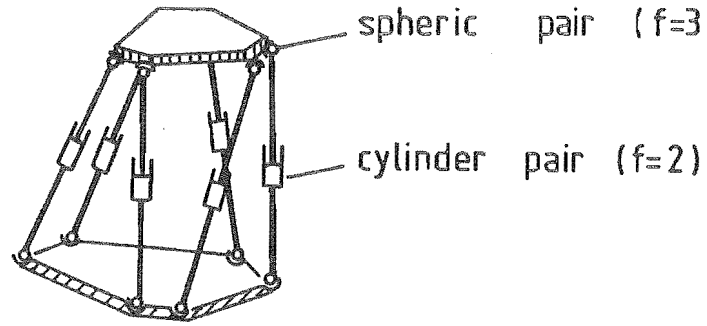


FIG 6·2 KINEMATIC MODEL OF THE DEVICE

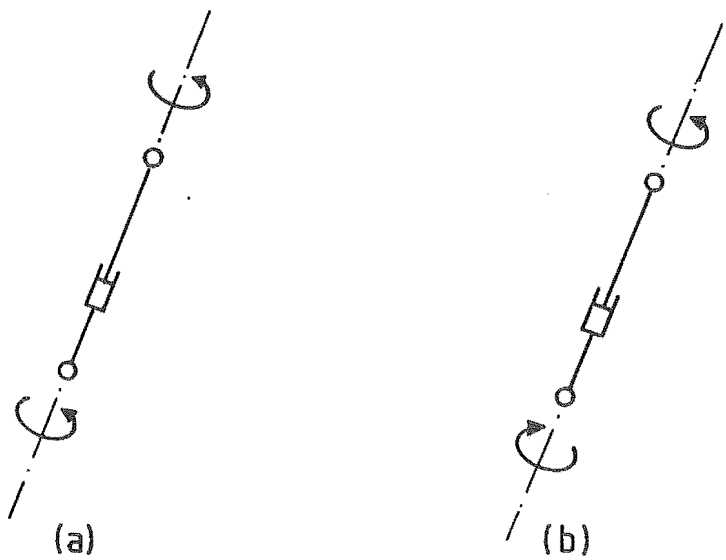


FIG 6·3 REDUNDANT FREEDOMS.

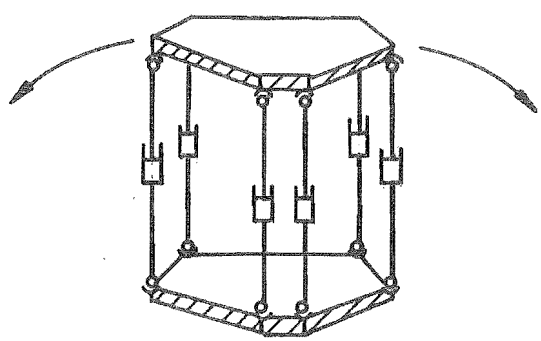


FIG 6·4 AN UNSTABLE CONFIGURATION

intersect at points A, B, C on the base and D, E, F on the platform as shown in Fig.6.5.

Let the displacements of the platform from its neutral location be represented by the vector

$$\underset{\sim}{\Delta d} = \begin{pmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}$$

x, y, and z being the translations and

θ_x, θ_y, and θ_z the rotations of the platform, along and about a triad of global coordinate axes ($\underset{\sim}{i}$, $\underset{\sim}{j}$, $\underset{\sim}{k}$).

Let the changes in the six ram lengths associated with these displacements be grouped as the vector

$$\underset{\sim}{\Delta \ell} = \begin{pmatrix} \Delta\ell_1 \\ \Delta\ell_2 \\ \vdots \\ \Delta\ell_6 \end{pmatrix}$$

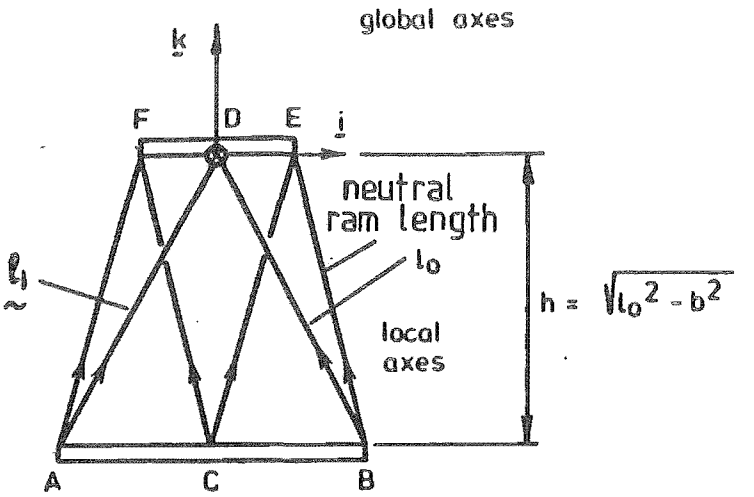
If the platform displacements and the ram length changes are small, their relationship can be expressed linearly as

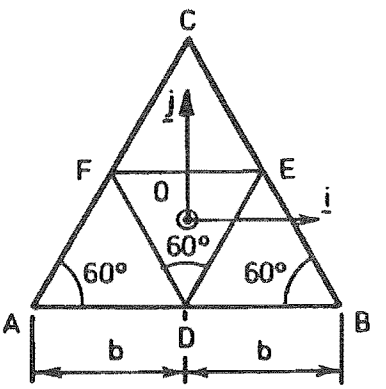
$$\underset{\sim}{\Delta \ell} = [a] \ \underset{\sim}{\Delta d}$$

[a] is the compatibility matrix (see McCallion [93]).

We shall derive the first row of [a] and then state the expressions for the complete matrix.

Referring to Figs 6.5 and 6.6, we can write





Coordinates about global axes:

A: $\left(-b, \frac{-b\sqrt{3}}{3}, -h\right)$

B: $\left(b, \frac{-b\sqrt{3}}{3}, -h\right)$

C: $\left(0, \frac{2b\sqrt{3}}{3}, -h\right)$

D: $\left(0, \frac{-b\sqrt{3}}{3}, 0\right)$

E: $\left(\frac{b}{2}, \frac{b\sqrt{3}}{6}, 0\right)$

F: $\left(\frac{-b}{2}, \frac{b\sqrt{3}}{6}, 0\right)$

FIG 6·5 SIMPLIFIED DEVICE GEOMETRY

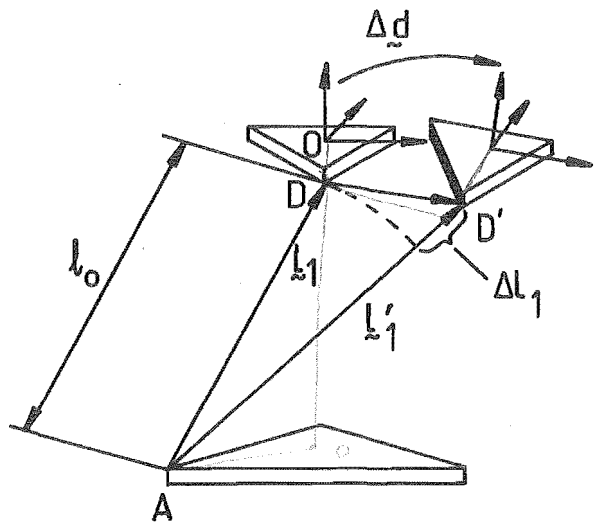


FIG 6·6 RAM VECTORS AND PLATFORM DISPLACEMENT

$$\Delta \ell_1 \simeq \frac{\underset{\sim}{\ell}_1 \cdot (\underset{\sim}{\ell}_1' - \underset{\sim}{\ell}_1)}{\ell_o}$$

where $\underset{\sim}{\ell}_1 = \underset{\sim}{OD} - \underset{\sim}{OA} = \begin{bmatrix} b \\ o \\ h \end{bmatrix}$

and $\underset{\sim}{\ell}_1' = \underset{\sim}{OD}' - \underset{\sim}{OA}$

$$\simeq \begin{pmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{pmatrix} \underset{\sim}{OD} + \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \underset{\sim}{OA}$$

$$\simeq \begin{pmatrix} \frac{b\sqrt{3}}{3}\theta_z + x + b \\ y \\ \frac{b\sqrt{3}}{-3}\theta_x + z + h \end{pmatrix}$$

$$\therefore \Delta \ell_1 = \frac{1}{\ell_o}\left(bx + hz - bh\frac{\sqrt{3}}{3}\theta_x + b^2\frac{\sqrt{3}}{3}\theta_z\right)$$

and the first row of [a] is

$$\begin{bmatrix} \frac{b}{\ell_o} & 0 & \frac{h}{\ell_o} & -\frac{bh\sqrt{3}}{3\ell_o} & 0 & \frac{b^2\sqrt{3}}{3\ell_o} \end{bmatrix}$$

By repeating the above procedure for links 2 to 6, we obtain the entire [a] matrix:

$$[a] = \frac{1}{\ell_o}\begin{bmatrix} b & 0 & h & \frac{-bh\sqrt{3}}{3} & 0 & \frac{b^2\sqrt{3}}{3} \\ -b & 0 & h & \frac{-bh\sqrt{3}}{3} & 0 & \frac{-b^2\sqrt{3}}{3} \\ \frac{-b}{2} & \frac{b\sqrt{3}}{2} & h & \frac{bh\sqrt{3}}{6} & \frac{-bh}{2} & \frac{b^2\sqrt{3}}{3} \\ \frac{b}{2} & \frac{-b\sqrt{3}}{2} & h & \frac{bh\sqrt{3}}{6} & \frac{-bh}{2} & \frac{-b^2\sqrt{3}}{3} \\ \frac{-b}{2} & \frac{-b\sqrt{3}}{2} & h & \frac{bh\sqrt{3}}{6} & \frac{bh}{2} & \frac{b^2\sqrt{3}}{3} \\ \frac{b}{2} & \frac{b\sqrt{3}}{2} & h & \frac{bh\sqrt{3}}{6} & \frac{bh}{2} & \frac{-b^2\sqrt{3}}{3} \end{bmatrix} \quad \text{Eq.6.1}$$

6.3 STRUCTURAL ANALYSIS

6.3.1 Local Stiffness Matrix [s]

If we assume that (i) friction in the spheric pairs is negligible and

the forces on the rams are purely axial, (ii) the springs in all the rams have equal stiffness k, and (iii) six local coordinate axes are chosen which are directed along the ram axes (Fig.6.5), then the local stiffness matrix relating ram forces to ram length changes is

$$[s] = k\,[I] \qquad \text{Eq. 6.2}$$

where $[I]$ is the 6 x 6 identity matrix.

6.3.2 Global Stiffness Matrix [S]

The global stiffness matrix [S], which relates the global platform forces to the global platform displacements, can be computed from the local stiffness matrix [s] as follows:

$$[S] = [a]^T\,[s]\,[a] \qquad \text{Eq. 6.3}$$

where $[a]^T$ is the transpose of the compatibility matrix [a] (again, see McCallion [93], for example).

From Eqs 6.1, 6.2 and 6.3

$$[S] = \frac{k}{\ell_o^2}\begin{bmatrix} 3b^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3b^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6h^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & b^2h^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & b^2h^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2b^4 \end{bmatrix}$$

6.3.3 Global Flexibility Matrix [C]

As [C] is the inverse of [S]

$$[C] = \frac{\ell_o^2}{k} \begin{bmatrix} \frac{1}{3b^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3b^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6h^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{b^2h^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\frac{1}{b^2h^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2b^4} \end{bmatrix}$$

6.4 PEG-HOLE INSERTION

The peg-hole insertion problem has been detailed elsewhere by McCallion et al [15]. In brief, insertion is the final and most difficult phase in the assembly of a peg and a block with a hole. Insertion begins when the peg enters the 'insertion funnel', an imaginary frustum whose angle is defined by the peg-hole diameter ratio (Fig.6.7).

During insertion, the peg can be linearly or angularly displaced relative to the hole (Figs 6.8a and 6.8b). Clearly, to correct a linear misalignment, a linear movement must be made, and to correct an angular misalignment, an angular movement must be made.

Fig.6.8a shows that in the case of a linear misalignment, there is a centering force acting on the peg, which tends to push it towards the hole. If the peg translates in the direction of this force, the linear misalignment will be corrected. Similarly, if the peg rotates in the direction of the righting moment due to an angular misalignment, the misalignment will be reduced.

We can readily appreciate that this ideal behaviour will be approached when the peg is connected to the insertion machine via the compliant device analysed in the previous sections (Fig.6.9) since the displacements of the

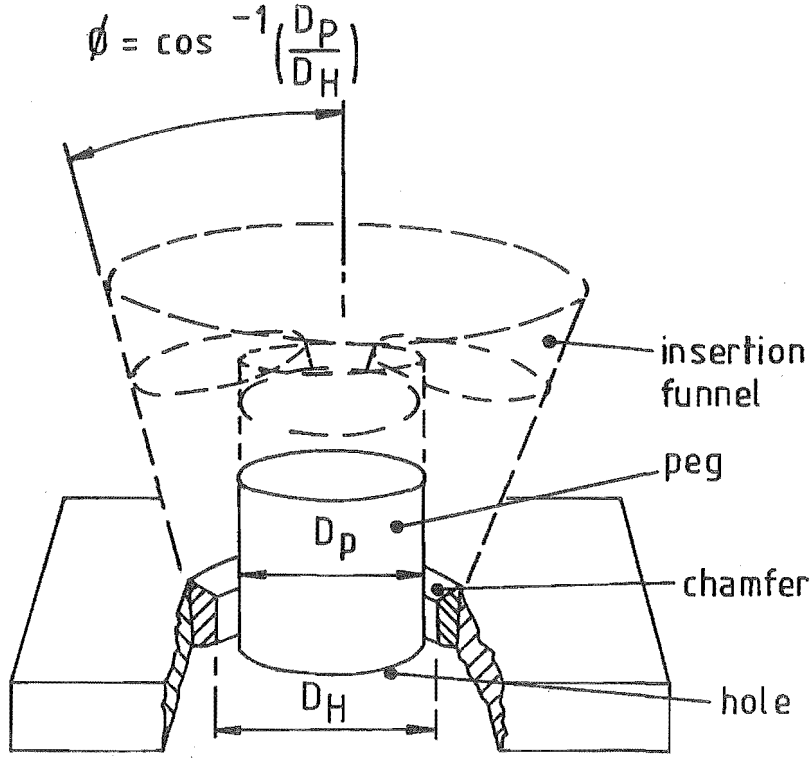
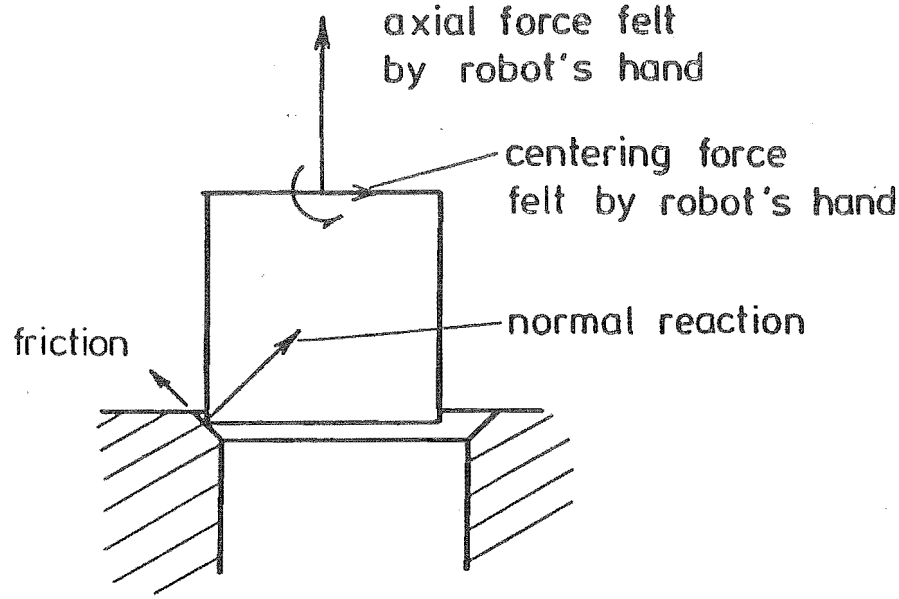
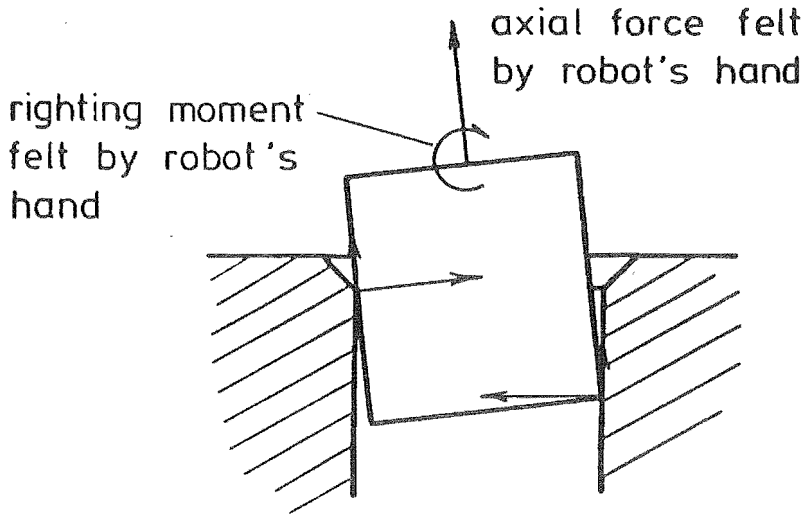


FIG 6·7 THE INSERTION FUNNEL



(a) linear misalignment



(b) angular misalignment

FIG. 6·8 COMPONENT INTERACTION DURING INSERTION.
(QUASI-STATIC CASE)

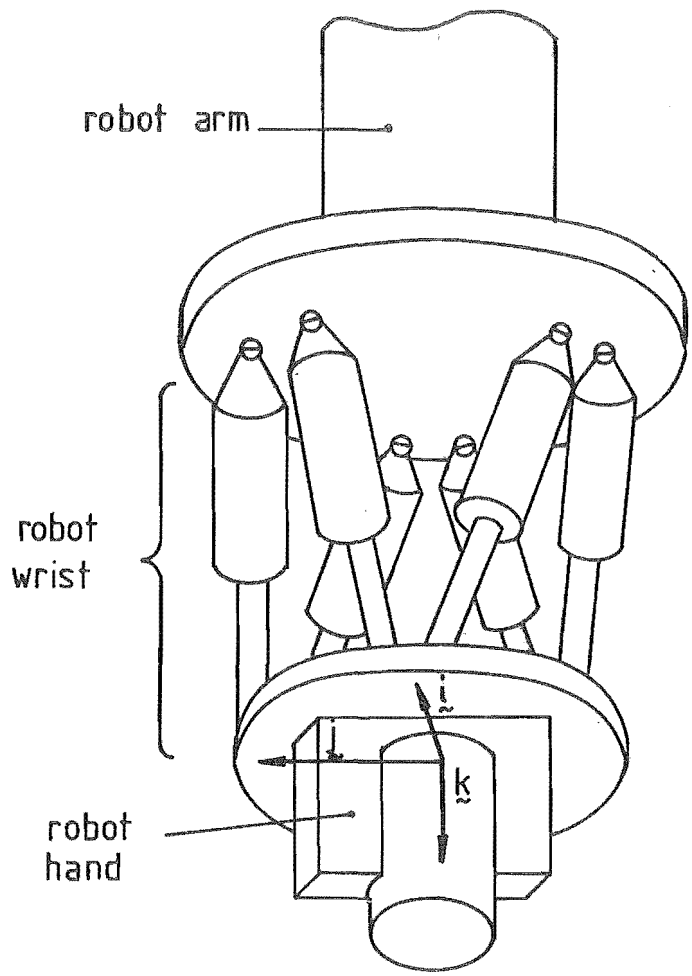


FIG 6·9 THE COMPLIANT DEVICE AS A ROBOT'S WRIST

peg will then be given by

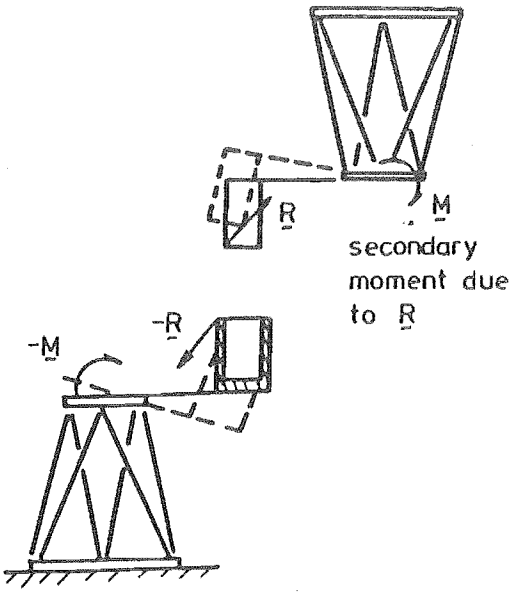
$$\begin{pmatrix} x \\ y \\ z \\ \theta x \\ \theta_y \\ \theta_z \end{pmatrix} = \frac{\ell_o^2}{k} \begin{pmatrix} \frac{1}{3b^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3b^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6h^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{b^2h^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{b^2h^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2b^4} \end{pmatrix} \begin{pmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{pmatrix}$$

where F_x, F_y, F_z, M_x, M_y, M_z are the reaction forces and moments on the peg resolved in $(\underset{\sim}{i}, \underset{\sim}{j}, \underset{\sim}{k})$. That is, the peg will be able to comply to the constraints imposed on it by the hole and will therefore automatically align itself with the hole.

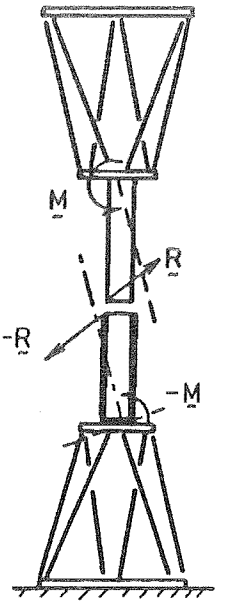
With the device mounted on a drill press as shown in Plate 6.1, we have inserted pegs of sizes ranging from 12 mm to 50 mm in diameter and 25 mm to 100 mm in length, into holes having diametral clearances ranging from 12 μm to 24μm, starting with misalignments between 1 mm to 2 mm and 1.5° to 2.5°.

These experiments have also been successfully repeated with the device fixed to the end-effector of a Unimate 2000B, an industrial robot whose inherent positional repeatability is only around ±1 mm (Plate 6.2).

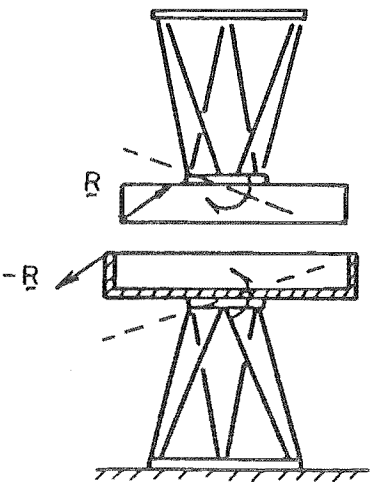
Note that all the pegs tried on the device were centrally located, relatively short, and of sufficiently small diameters. For off-centred or long pegs, the arrangements shown in Figs 6.10a and 6.10b will theoretically counteract the effects of the secondary moments due to the reaction forces on the tip of the pegs. These arrangements will not work with pegs having very large diameters (Fig.6.10c). In these cases, the pegs should be designed so that the reaction forces pass through the origin of $(\underset{\sim}{i}, \underset{\sim}{j}, \underset{\sim}{k})$.



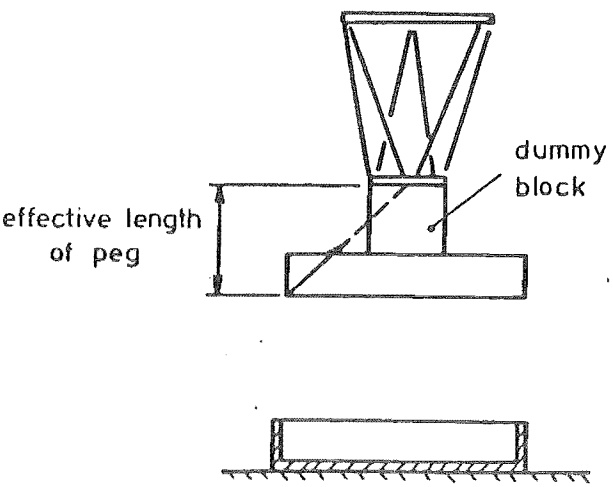
(a) off-centered peg and hole



(b) long peg and hole

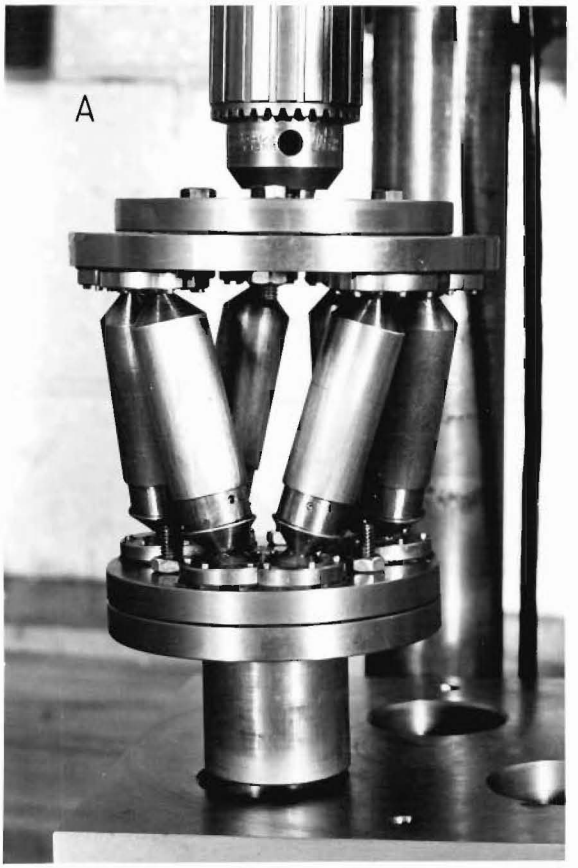


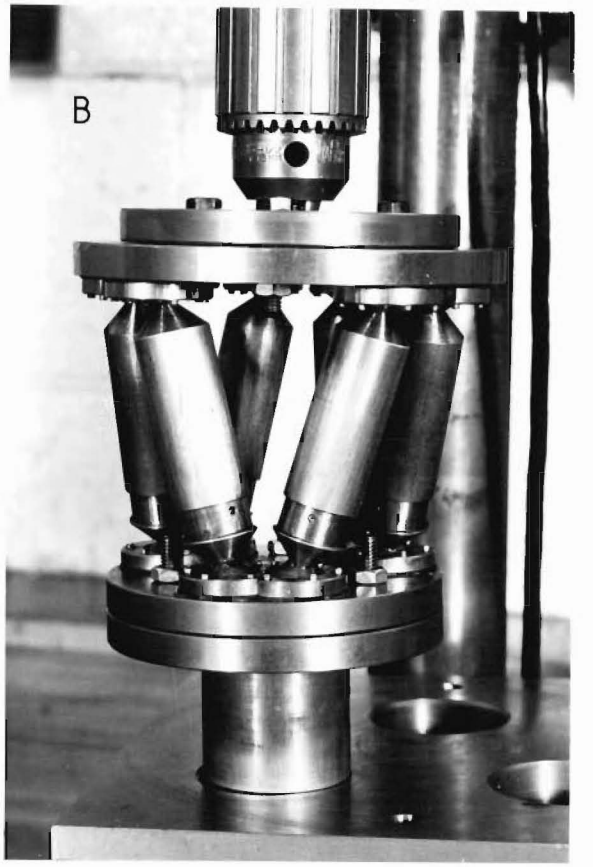
(c) peg and hole with large diameters



(d) correct arrangement when peg and hole have large diameters

FIG. 6·10 COUNTERACTING SECONDARY MOMENTS





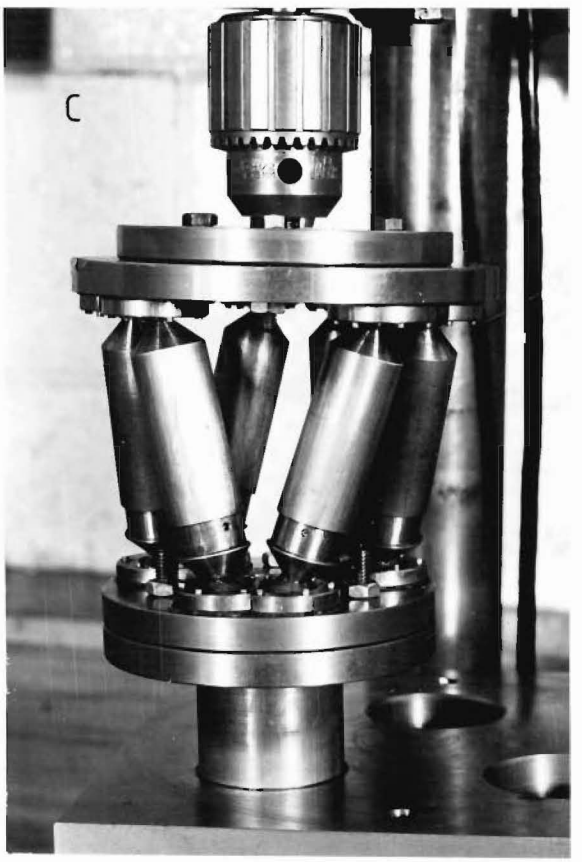


PLATE. 6·1 THE INSERTION OF A 50·00 mm ϕ PEG INTO A 50·02 mm ϕ HOLE

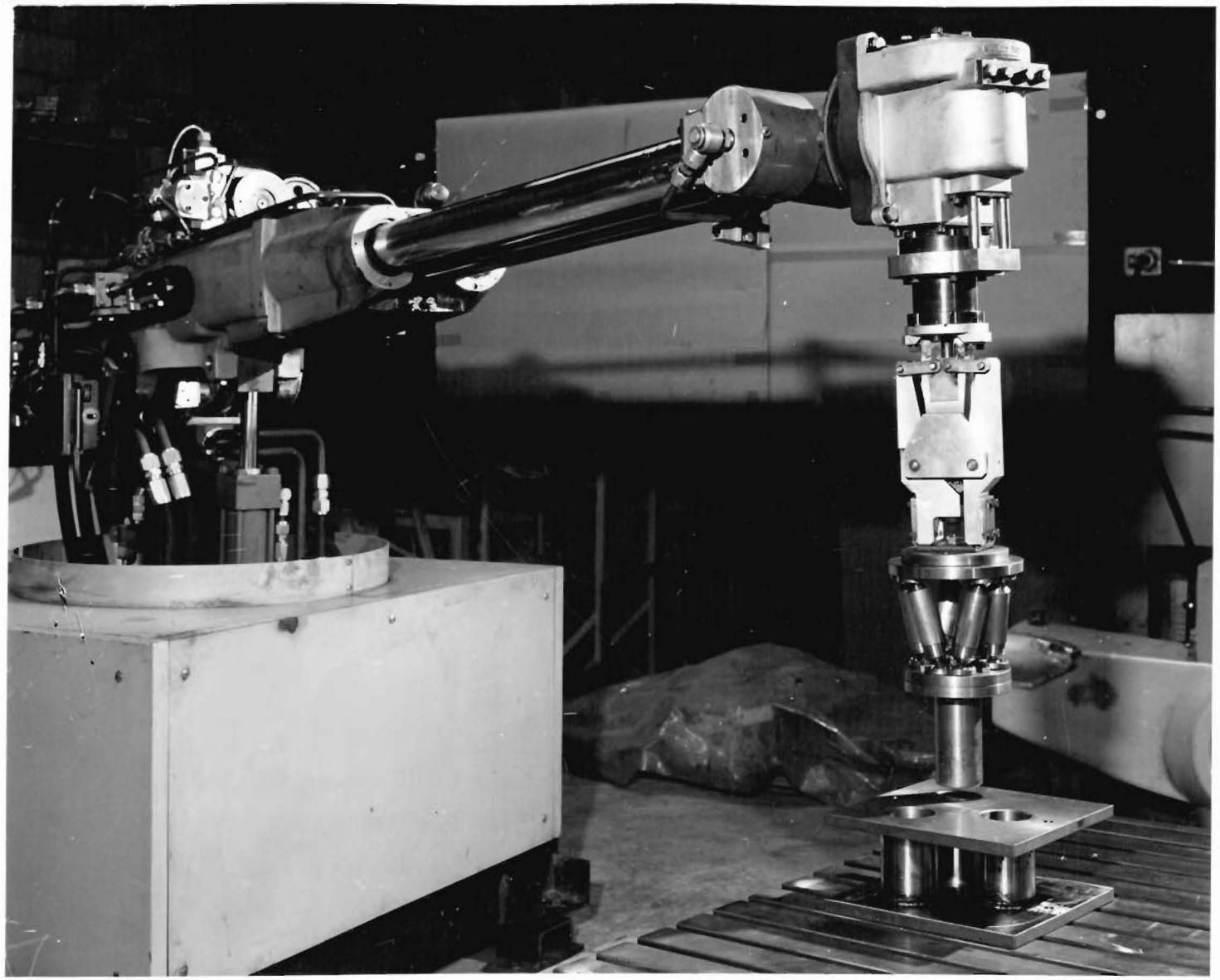


PLATE 6·2 THE COMPLIANT DEVICE ON A UNIMATE 2000B

By courtesy of the D.S.I.R., Lower Hutt

This means, if the chamfer angle is 45°, and friction is small, the effective length-to-diameter ratio of the pegs should be approximately equal to 0.5 (Fig.6.10d).

6.5 DISCUSSION

Obviously, there are many possible variations to the construction of our compliant device. For example, instead of being spring-loaded, the rams can be pneumatically or hydraulically actuated. Linear Variable Differential Transformers (LVDT's) mounted along the rams will provide accurate feedback of the platform-base location as well as some measure of the reaction forces and moments on the peg. If these are masked by excessive friction in the ball joints, the latter can be replaced by universal joints, provided that one half of each ram can spin freely relative to the other about the ram's axis. Another form of compliant device could be a suitably moulded block of rubber. Finite-element analysis combined with experimentation would be required to yield the form of the moulding to give optimal force-displacement characteristics.

6.6 SUMMARY

A simple device has been described which is capable of making purely translational motions in the direction of forces applied to it, and purely rotational motions in the direction of moments applied to it. The use of the device as a flexible wrist in a low cost, low precision robot for inserting pegs into holes with extremely small clearances has been discussed.

CHAPTER 7

CONCLUSION

This study has embraced the analysis, design and development of both active and passive aids aimed at extending the scope of industrial robots to the final aligning and insertion phases in the assembly of discrete mechanical components.

Initially, a survey of the general field of assembly mechanisation was conducted with a view to uncovering potential aids. Several of these were found. They were mainly of the active type, requiring the conscious application of either visual or tactile feedback, or both. It was then theoretically established that none of these aids were feasible in their existing form, visual systems being slow and inaccurate, and tactile systems probabilistic and inflexible.

The probabilistic nature of tactile systems, which was evident from their reliance on blind search procedures, was attributed to the information deficiency characteristic of traditional tactile feedback.

Traditionally, all tactile feedback schemes involved a single sensor array coupled either to a component-placing mechanism (e.g. a robot arm) or to a work-holding fixture (located on some work table). It was contended that the information deficiency would be remedied if two sensor arrays were used, one coupled to the mechanism and the other to the fixture. Various deterministic schemes based on this bilateral arrangement were proposed. Only touch and force sensors were considered although the basic principles involved would equally be applicable to combinations of other sensors.

Next, the kinematics of an assembly machine capable of implementing the proposed bilateral feedback schemes were investigated. The machine, a

computer-driven six-degree-of-freedom linkage, had been built on the premise that it would assist a high-speed assembly robot with the fine and accurate movements during the detection and correction of the robot's positioning errors.

The kinematic investigation yielded control equations for the machine. Programs in Fortran and Assembly language (see Appendix A) were subsequently written to enable a mini-computer to solve these equations and to communicate the solutions to the electronic circuitry that actuates the drive motors in the machine (see Appendix B, Fig. 7.1, and Plate 7.1).

Both off-line and on-line tests were carried out to establish the accuracy of the programs, the correct performance of the electronics, and the integrity of the software-hardware complex. However, a full calibration exercise remains to be done for the assembly machine. Steps towards achieving this have already been taken [78].

Although in theory the proposed bilateral feedback systems would enable the assembly machine to eliminate all of the positioning errors committed by a pick-and-place robot, in practice this would not be the case due to the finite resolution of real sensors and imperfections within the assembly machine (e.g., clearances in its moving parts and deflection of its structural members). Thus, typically this consortium of *active* assembly aids would only be suitable for the aligning phase. The final section of this study was devoted to the analysis of a novel *passive* aid ideal for the insertion phase.

The device, also with six degrees of freedom, was shown to be capable of making purely translational, or purely rotational, displacements in the direction of any forces, or moments, applied to it. This susceptibility to external constraints accounted for the device's tendency passively to correct the misalignment of a component during its insertion into its mating part. The true effectiveness of the device was highlighted on a common industrial

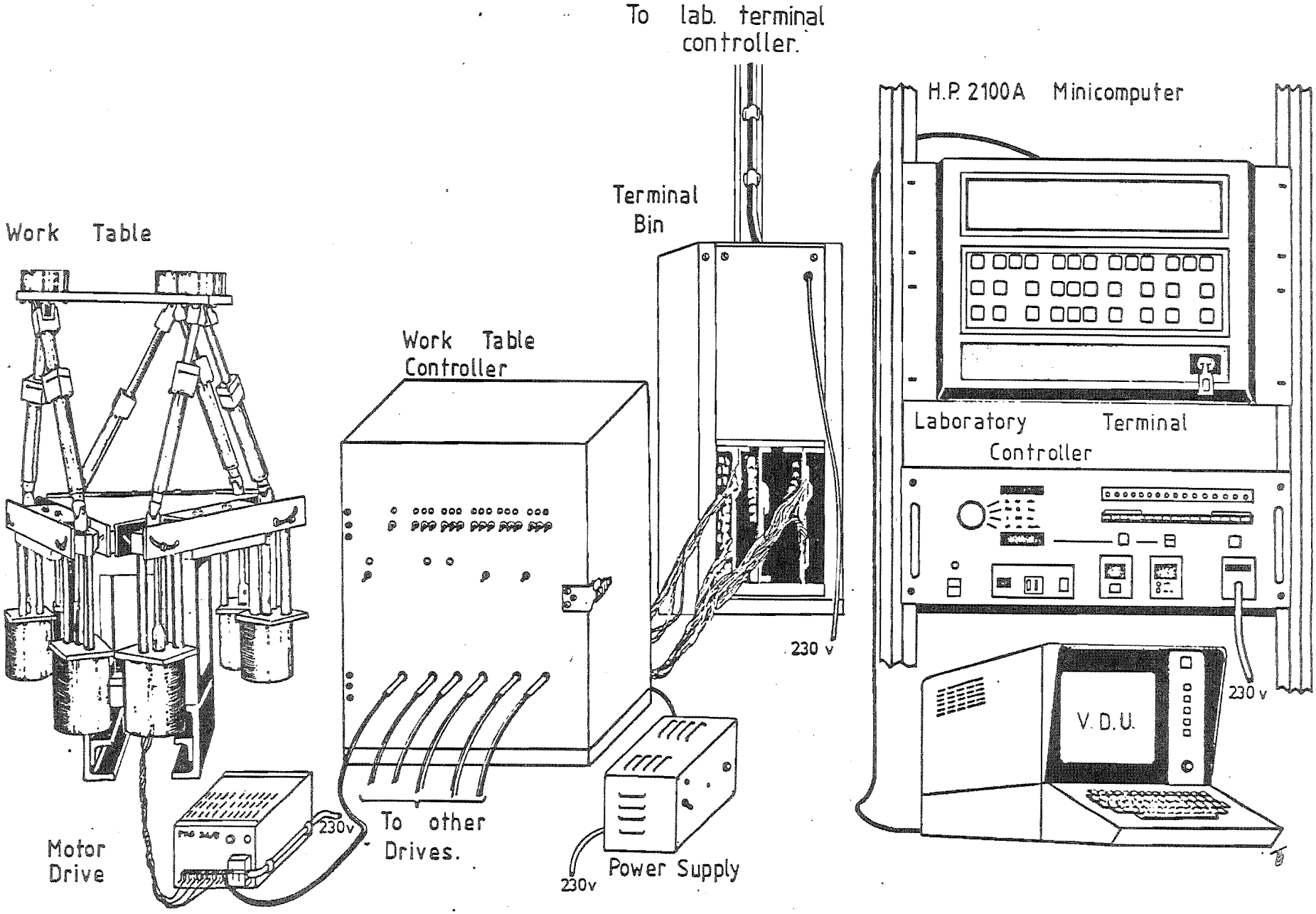


FIG 7-1 A SKETCH OF THE COMPLETE TEST RIG

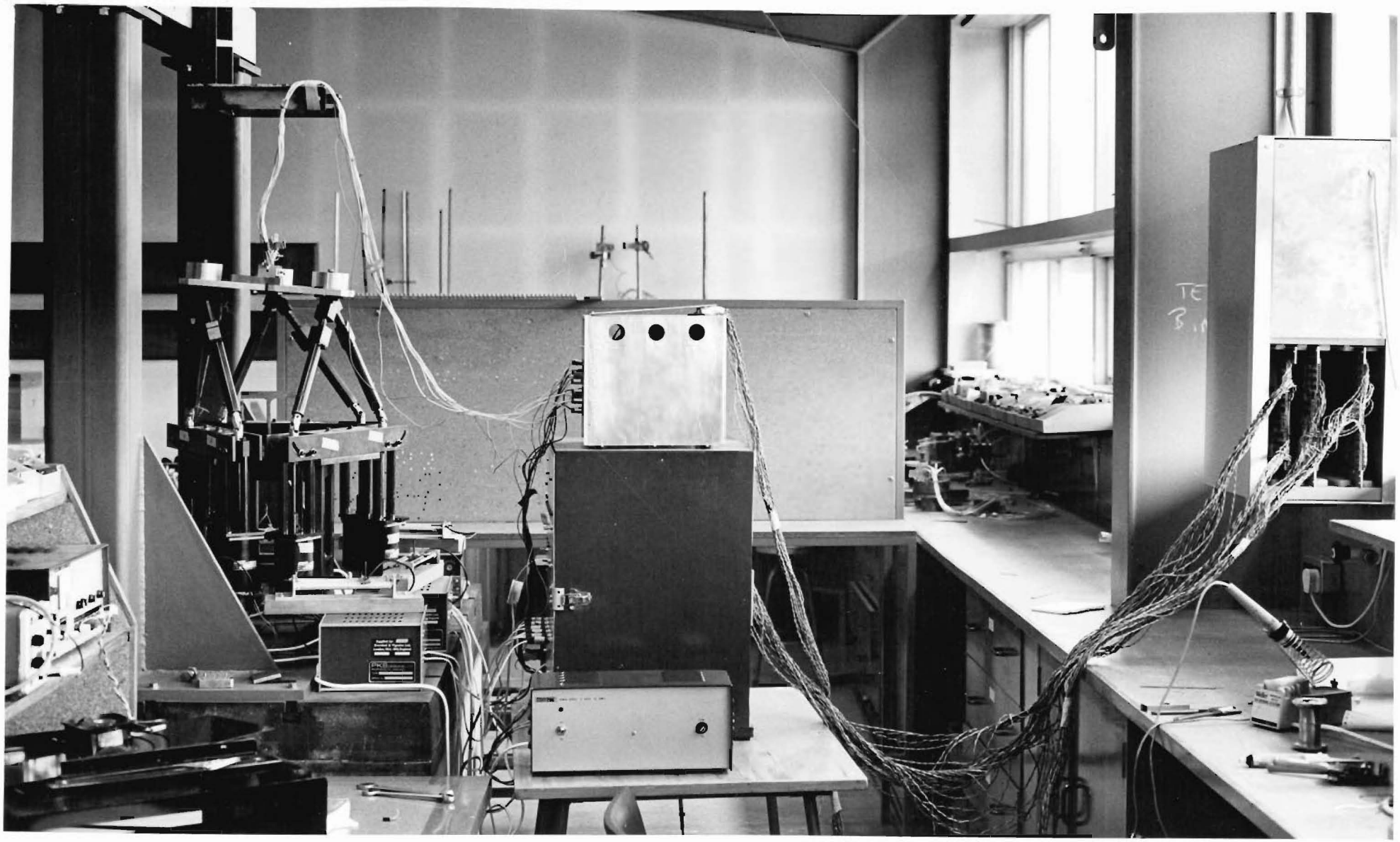


PLATE 7·1 THE COMPLETE TEST RIG (Remote computer not shown)

robot: with the device mounted in its end effector, the robot could insert cylindrical pegs into cylindrical holes (and twin pegs into twin holes) with clearances fifty times smaller than the robot's inherent positional repeatability.

APPENDIX A

SOFTWARE FOR CONTROLLING THE WORK TABLE

```
0001   FTN4
0002   C  PROGRAM TO COMMUNICATE WITH OPERATOR
0003   C  PHAM D.T.
0004   C  SEVENTH VERSION
0005   C  PURPOSE: TO INPUT PLATFORM TRAJECTORY AND OUTPUT REQUIRED
0006   C            LEG LENGTHS
0007   C           AND TOTAL NO. OF MOTOR STEPS
0008   C           AND SPEED NUMBERS
0009   C  THIS VERSION USES RODRIGUES PARAMETERS .
0010   C  THE PARAMETERS ARE WELL BEHAVED AND EASY TO VISUALISE
0011   C  2/04/1978
0012   C  20/04/1978  MINOR MISTAKES CORRECTED ( SEE LINES 259,261 )
0013   C  02/08/1979 SUBROUTINE START MODIFIED
0014   C  GLOSSARY OF ERROR MESSAGES
0015   C    ERROR 1 : IN THE TABLE CONFIGURATION TO BE ANALYSED BY CORD4
0016   C              ,THE LEG LENGTHS ARE LESS THAN 300MM OR GREATER
0017   C               THAN 350MM(FROM CORD4).
0018   C    ERROR 2 : MAXIT EXCEEDED WITHOUT CONVERGENCE(FROM NLNEQ).
0019   C    ERROR 3 : SINGULAR JACOBIAN ( FROM NLNEQ)
0020   C    ERROR 4 : NUMBER OF TRAJECTORY POINTS LESS THAN OR EQUAL
0021   C              TO 0 ( FROM OPER,ON COMING BACK FROM INTRA).
0022   C    ERROR 5 : NMEGA GREATER THAN -70(OR ABOUT 300 STEPS/SEC)
0023   C              OR NSTEP LESS THAN 0 OR GREATER THAN 3000(FROM OPER)
0024   C              OR LEG LENGTHS LESS THAN ORIGINAL LEG LENGTHS
0025   C              (FROM NULEG).
0026   C    ERROR 6 : AN INTERRUPT OCCURRED ON OR BEFORE THE 7TH DATA
0027   C              STROBE.SCANNING MUST BE REPEATED(FROM OMPLX).
0028   C    ERROR 7 : THE AWAITED 8TH-DATA-STROBE INTERRUPT DOES NOT
0029   C               OCCUR,EVEN THOUGH 7 ATTEMPTS HAVE BEEN MADE
0030   C               (FROM OMPLX).
0031   C              POSSIBILITIES:MULTIPLEXER BOARD MALFUNCTIONS
0032   C                            OR ENCOD BOARD IS NOT WORKING,
0033   C                            OR TRANSFER REQUEST LINE IS STUCK
0034   C                            OR THE DATA ACQUISITION SYSTEM
0035   C                            HAS PACKED IN.
0036   C    ERROR 8 : INTERRUPT OCCURRED ON THE 8TH DATA STROBE BUT
0037   C              IT DOES NOT COME FROM THE CURRENTLY ADDRESSED
0038   C              SUB-DEVICE(FROM OMPLX).
0039          PROGRAM OPER1
0040          EXTERNAL OMPLX,IDENT,ICLR,WPANL,INITL
0041          DOUBLE PRECISION DGLX(3),DGLY(3),DGLZ(3)
0042         1,NGLX,NGLY,NGLZ
0043         1,GENRL(6),A(3,3)
0044          REAL B(3),TOP(3),SQTOP(3),ALEG(6),GAMMA(3)
0045         1,AGLX(3),AGLY(3),AGLZ(3),BGLX(3),BGLY(3),BGLZ(3)
0046         1,ABX(3),ABY(3),ABZ(3)
0047         1,MOD
0048         1,EPX(3),EPY(3),EPZ(3)
0049         1,R(3,3),VECT(3)
0050         1,TEMP(6)
0051         1,COORD(600)
0052          INTEGER IWRIT,ERROR
0053         1,NPOIN
0054         1,NSTEP(6)
0055         1,ISTEP(6),MSTEP(6),NMEGA(6)
0056          COMMON IDEV,IDATA(6),JFAIL,IFLAG
0057          COMMON P(3),Q(3),COSGA(3),SINGA(3),U(3),V(3),C(3),H(3)
0058         1,SQTOP,B,GAMMA,TOP
0059         1,DGLX,DGLY,DGLZ,NGLX,NGLY,NGLZ
0060         1,EPX,EPY,EPZ
0061         1,MOD
0062         1,ALEG
0063         1,GENRL
0064         1,A
0065         1,ERROR
0066         1,IWRIT
0067         1,AGLX,AGLY,AGLZ,BGLX,BGLY,BGLZ,ABX,ABY,ABZ
0068         1,UORGX(6),UORGY(6),UORGZ(6),ALEG0(6)
0069         1,AMOTX(6),AMOTY(6),AMOTZ(6),PITCH
0070   C  CLOCK FREQUENCY
```

```
0071          CLOCK = 42300.
0072    C   INITIALISATION
0073          CALL DTABL
0074          CALL START
0075          IFRST = 1
0076    C   READ ORIGINAL LEG LENGTHS AND COMPUTE ORIGINAL COORDINATES
0077    1     ERROR = 0
0078          WRITE(1,11)
0079    11    FORMAT(" ENTER 6 LEG LENGTHS,EACH BETWEEN 300 AND 350 MM")
0080          READ(1,*)ALEG
0081          IWRIT = 0
0082          CALL CORD4
0083          IF(ERROR.NE.0) GO TO 30
0084    C   NUMBER OF STEPS AT REFERENCE POSITION
0085          L = 0
0086          WRITE(6,55)L
0087          DO 90 I=1,6
0088    90     TEMP(I) = GENRL(I)
0089          WRITE(6,66)TEMP
0090          CALL TR(TEMP,R,VECT)
0091          IWRIT = 1
0092          CALL NULEG(R,VECT,NSTEP)
0093           IF(ERROR.NE.0) GO TO 30
0094          DO 95 I=1,6
0095           ISTEP(I) = NSTEP(I)
0096           NSTEP(I) = 0
0097    95    CONTINUE
0098    C   INPUT TRAJECTORY
0099    2     CALL INTRA(NPOIN,COORD)
0100          IF(NPOIN.GT.0)GO TO 25
0101          ERROR = 4
0102          GO TO 30
0103    25    WRITE(1,24)
0104    24    FORMAT(" HARD COPY OF POSITION AND SPEED NUMBERS REQUIRED?
0105         1 YES = 1,NO = 0")
0106          READ(1,*)IWRIT
0107          WRITE(1,26)
0108    26    FORMAT(" ENTER TIME BETWEEN LOCATIONS,IN SECONDS")
0109          READ(1,*) TIME
0110    C   COMPUTE LEG LENGTHS AT ALL TRAJECTORY POINTS
0111          INDEX = 1
0112          DO 200 I=1,NPOIN
0113    C   GET NEW COORDINATES AND STORE OLD POSITION NUMBERS
0114           DO 100 J=1,6
0115            TEMP(J) = COORD(INDEX)
0116            INDEX = INDEX+1
0117           MSTEP(J) = NSTEP(J)
0118    100    CONTINUE
0119           IF(IWRIT.EQ.0) GO TO 101
0120           WRITE(6,55)I
0121    55     FORMAT(" LOCATION NO.",I5)
0122           WRITE(6,66)TEMP
0123    66     FORMAT(3E20.10)
0124    101    CALL TR(TEMP,R,VECT)
0125           CALL NULEG(R,VECT,NSTEP)
0126            IF(ERROR.NE.0) GO TO 30
0127    C   OUTPUT POSITION AND SPEED NUMBERS
0128          DO 110 K=1,6
0129           NSTEP(K) = NSTEP(K)-ISTEP(K)
0130           OMEGA = ABS((NSTEP(K)-MSTEP(K))/TIME)
0131    C   SET MINIMUM SPEED = 1 STEP/SEC
0132           IF(OMEGA.LT.1.) OMEGA = 1.
0133           NMEGA(K) = -CLOCK/(2*OMEGA)
0134    C   SAFETY MEASURES : NSTEP MUST BE INSIDE ALLOWABLE RANGE
0135    C                     NMEGA CANNOT BE TOO LARGE
0136            IF((NSTEP(K).LT.0).OR.(NSTEP(K).GT.3000)) ERROR = 5
0137           IF(NMEGA(K).GT.-70) ERROR = 5
0138    110   CONTINUE
0139            IF(ERROR.NE.0) GO TO 30
0140          IF(IWRIT.EQ.0)GO TO 112
```

```
0141          WRITE(6,77)
0142    77    FORMAT(" POSITION NUMBERS:")
0143          WRITE(6,99) NSTEP
0144          WRITE(6,88)
0145    88    FORMAT(" SPEED NUMBERS:")
0146          WRITE(6,99) NMEGA
0147    99    FORMAT(6I10)
0148    C  OUTPUT SPEED NUMBERS TO 6DOF TABLE
0149    112   DO 150 L=1,6
0150    150     IDATA(L) = NMEGA(L)
0151           IDEV  = 1
0152           JFAIL = 0
0153          CALL OMPLX
0154          IF(JFAIL.NE.0) GO TO 190
0155          CALL ICLR
0156    C  OUTPUT POSITION NUMBERS TO 6DOF TABLE
0157          DO 160 L=1,6
0158    160    IDATA(L) = NSTEP(L)
0159          IDEV = 2
0160          JFAIL = 0
0161          CALL OMPLX
0162          IF (JFAIL.NE.0) GO TO 190
0163          CALL ICLR
0164    C  CHECK FOR FIRST ROUND
0165          IF(IFRST.EQ.0) GO TO 170
0166          IFRST = 0
0167          IOK   = 1
0168          CALL WPANL(IOK)
0169    170   IF(IFLAG.EQ.4) GO TO 180
0170          GO TO 170
0171    180   CALL ICLR
0172          DUMMY = 0
0173          DUMMY = 0
0174          CALL WPANL(I)
0175    200   CONTINUE
0176          GO TO 45
0177    190   IERR  = JFAIL-512
0178          ERROR = JFAIL+5
0179    30    WRITE(1,22) ERROR
0180    22    FORMAT(" ERROR NO.",I3)
0181    40    WRITE(1,33)
0182    33    FORMAT(" TRY AGAIN? YES=1,NO=0")
0183          READ(1,*)IANS
0184          IF(IANS.EQ.1) GO TO 1
0185          GO TO 46
0186    45    WRITE(1,44)
0187    44    FORMAT(" MORE POINTS? YES =1,NO =0")
0188          READ(1,*)IANS
0189          IF(IANS.EQ.1) GO TO 2
0190    46    STOP
0191          END
```

```
0192  C
0193  C    SUBROUTINE START :TO INITIALISE ROBOT HARDWARE
0194  C
0195          SUBROUTINE START
0196          EXTERNAL ICLR,INITL
0197          CALL INITL
0198          CALL ICLR
0199          RETURN
0200          END
```

```
0201   C
0202   C   SUBROUTINE INTRA
0203   C   INPUT TRAJECTORY OF 6-DOF TABLE
0204   C   21/03/1978
0205   C   SECOND VERSION
0206   C
0207         SUBROUTINE INTRA( NPOIN,COORD )
0208         EXTERNAL EXEC
0209         INTEGER NPOIN
0210         REAL COORD(600),TEMP(6)
0211         ND = 0
0212         DO 50 I= 1,6
0213   50    TEMP(I) = 0.
0214         INDEX = 1
0215   5      WRITE(1,21)
0216   21     FORMAT(" ENTER 1 FOR VDU INPUT"/"    OR 5 FOR TAPE INPUT")
0217          READ(1,*)LU
0218          IF(LU.EQ.5)GO TO 10
0219          IF(LU.NE.1) GO TO 5
0220         WRITE(1,11)
0221   11    FORMAT(" INPUT TRAJECTORY OF PLATFORM AS A SEQUENCE OF GEN."/
0222        1" COORDINATES ."/
0223        1" FORMAT : ND,COORD(1),....,COORD(6) "/
0224        1" ND = A BLANK FOR NORMAL INPUT"/
0225        1" ND = 1 FOR TERMINATING INPUT"/)
0226   C  INPUT LOOP
0227   10    DO 200 I=1,101
0228          IF(I.EQ.101) GO TO 200
0229   12     IF(LU.EQ.5) GO TO 15
0230          WRITE(1,22) I
0231   22     FORMAT(" ENTER COORDINATE SET NO.",I5)
0232   15     READ(LU,*) ND,TEMP
0233   C  CHECK END OF INPUT(ND=1)
0234          IF(ND.EQ.1)GO TO 30
0235   C  CHECK VALIDITY OF INPUT
0236          IF((TEMP(1).GT.-60.).AND.(TEMP(1).LT.60.).AND
0237        1   .(TEMP(2).GT.-60.).AND.(TEMP(2).LT.60.).AND
0238        1   .(TEMP(4).GT.-200.).AND.(TEMP(4).LT.200.).AND
0239        1   .(TEMP(5).GT.-200.).AND.(TEMP(5).LT.200.).AND
0240        1   .(TEMP(6).GT.-200.).AND.(TEMP(6).LT.200.).AND
0241        1   .(TEMP(3).GT.260.).AND.(TEMP(3).LT.320.))  GO TO 20
0242          WRITE(1,33)
0243   33     FORMAT(" INPUT ERROR")
0244          GO TO 12
0245   C  STORE INPUT IN COORDINATE ARRAY
0246   20    DO 100 I1=1,6
0247           COORD(INDEX) = TEMP(I1)
0248           IF(I1.GT.3) COORD(INDEX) = (COORD(INDEX))/1000.
0249           INDEX = INDEX+1
0250   100    CONTINUE
0251   200    CONTINUE
0252   30     NPOIN = I-1
0253         IF((NPOIN.LT.1).OR.(LU.EQ.5)) GO TO 35
0254   C  PUNCH PAPER TAPE
0255         CALL EXEC(3,1004B)
0256         K = 1
0257         DO 300 J=1,NPOIN
0258          ND = 0
0259          COO4 = COORD(K+3)*1000.
0260          COO5 = COORD(K+4)*1000.
0261          COO6 = COORD(K+5)*1000.
0262          WRITE(4,34) ND,COORD(K),COORD(K+1),COORD(K+2)
0263        1                ,COO4    ,COO5    ,COO6
0264   34     FORMAT(I2,",",F11.4,",",F11.4,",",F11.4,",",
0265        1               ,F11.4,",",F11.4,",",F11.4)
0266          K = K+6
0267   300    CONTINUE
0268         ND = 1
0269         WRITE(4,36)ND
0270   36    FORMAT(I2)
0271         CALL EXEC(3,1004B)
0272   35    RETURN
0273         END
```

```
0274  C
0275  C  SUBROUTINE DTABL
0276  C  SET TABLE PARAMETERS
0277         SUBROUTINE DTABL
0278         DOUBLE PRECISION DGLX(3),DGLY(3),DGLZ(3)
0279        1,NGLX,NGLY,NGLZ
0280        1,GENRL(6),A(3,3)
0281         REAL B(3),TOP(3),SQTOP(3),ALEG(6),GAMMA(3)
0282        1,AGLX(3),AGLY(3),AGLZ(3),BGLX(3),BGLY(3),BGLZ(3)
0283        1,ABX(3),ABY(3),ABZ(3)
0284        1,MOD
0285        1,EPX(3),EPY(3),EPZ(3)
0286         INTEGER IWRIT,ERROR
0287         COMMON IDEV,IDATA(6),JFAIL,IFLAG
0288         COMMON P(3),Q(3),COSGA(3),SINGA(3),U(3),V(3),C(3),H(3)
0289        1,SQTOP,B,GAMMA,TOP
0290        1,DGLX,DGLY,DGLZ,NGLX,NGLY,NGLZ
0291        1,EPX,EPY,EPZ
0292        1,MOD
0293        1,ALEG
0294        1,GENRL
0295        1,A
0296        1,ERROR
0297        1,IWRIT
0298        1,AGLX,AGLY,AGLZ,BGLX,BGLY,BGLZ,ABX,ABY,ABZ
0299        1,UORGX(6),UORGY(6),UORGZ(6),ALEG0(6)
0300        1,AMOTX(6),AMOTY(6),AMOTZ(6),PITCH
0301         P(1) = -155.
0302         P(2) = 211.5
0303         P(3) = -56.5
0304         Q(1) = -154.7298721
0305         Q(2) = -56.86900151
0306         Q(3) = 211.5988736
0307         B(1)= 310.
0308         B(2) = 310.
0309         B(3) = 310.
0310         GAMMA(1) = 0.
0311         GAMMA(2) = 120.
0312         GAMMA(3) = 240.
0313         TOP(1)   = 264.
0314         TOP(2)   = 264.
0315         TOP(3)   = 264.
0316         U(1)     = 5.
0317         U(2)     = 5.
0318         U(3)     = 5.
0319         V(1)     = 15.
0320         V(2)     = 15.
0321         V(3)     = 15.
0322  C  COORDINATES OF ORIGINAL MOTOR UNIVERSAL AXES (REFERENCE)
0323         UORGX(1) = 1.
0324         UORGY(1) = 0.
0325         UORGZ(1) = 0.
0326         UORGX(2) = 1.
0327         UORGY(2) = 0.
0328         UORGZ(2) = 0.
0329         UORGX(3) =-.5
0330         UORGY(3) = .8660254
0331         UORGZ(3) = 0.
0332         UORGX(4) =-.5
0333         UORGY(4) = .8660254
0334         UORGZ(4) = 0.
0335         UORGX(5) =-.5
0336         UORGY(5) =-.8660254
0337         UORGZ(5) = 0.
0338         UORGX(6) =-.5
0339         UORGY(6) =-.8660254
0340         UORGZ(6) = 0.
0341  C  MOTOR AXES
0342         AMOTX(1) = 0.
0343         AMOTY(1) = 0.
```

```
0344          AMOTZ(1) = 1.
0345          AMOTX(2) = 0.
0346          AMOTY(2) = 0.
0347          AMOTZ(2) = 1.
0348          AMOTX(3) = 0.
0349          AMOTY(3) = 0.
0350          AMOTZ(3) = 1.
0351          AMOTX(4) = 0.
0352          AMOTY(4) = 0.
0353          AMOTZ(4) = 1.
0354          AMOTX(5) = 0.
0355          AMOTY(5) = 0.
0356          AMOTZ(5) = 1.
0357          AMOTX(6) = 0.
0358          AMOTY(6) = 0.
0359          AMOTZ(6) = 1.
0360    C  ORIGINAL LEG LENGTHS
0361          ALEG0(1) = 300.
0362          ALEG0(2) = 300.
0363          ALEG0(3) = 300.
0364          ALEG0(4) = 300.
0365          ALEG0(5) = 300.
0366          ALEG0(6) = 300.
0367    C PITCH OF SCREWS , IN MM
0368          PITCH = 2.54
0369    C  PLATFORM COORDINATES OF CORNERS E
0370          AMEDI = SQRT(.5*(TOP(1)**2+TOP(2)**2-.5*TOP(3)**2))
0371          COSIN = (.75*TOP(2)+.25*TOP(1)*TOP(1)/TOP(2)
0372         1                    -.25*TOP(3)*TOP(3)/TOP(2))/AMEDI
0373          EPX(2) = .6666667*AMEDI*COSIN
0374          EPY(2) = .6666667*AMEDI*SQRT(1.-COSIN**2)
0375          EPX(3) = -TOP(2)+EPX(2)
0376          EPY(3) = EPY(2)
0377          EPX(1) = -EPX(2)-EPX(3)
0378          EPY(1) = -EPY(2)-EPY(3)
0379          EPZ(1) = 0.
0380          EPZ(2) = 0.
0381          EPZ(3) = 0.
0382    C
0383          DO 10 I=1,3
0384           AGLX(I) = P(I)
0385           AGLY(I) = Q(I)
0386           AGLZ(I) = 0.
0387           BGLX(I) = P(I)+B(I)*COS(.01745329252*GAMMA(I))
0388           BGLY(I) = Q(I)+B(I)*SIN(.01745329252*GAMMA(I))
0389           BGLZ(I) = 0.
0390           ABX(I)  = BGLX(I)-AGLX(I)
0391           ABY(I)  = BGLY(I)-AGLY(I)
0392           ABZ(I)  = 0.
0393    10     CONTINUE
0394          RETURN
0395          END
```

```
0396  C
0397  C  SUBROUTINE TR TO COMPUTE VECT T AND MATRIX R FROM GEN COORD
0398        SUBROUTINE TR(TEMP,R,VECT)
0399        REAL TEMP(6),R(3,3),VECT(3)
0400  C
0401        ALMDA = TEMP(4)
0402        AMU   = TEMP(5)
0403        ANU   = TEMP(6)
0404        SQLMD = ALMDA**2
0405        SQMU  = AMU**2
0406        SQNU  = ANU**2
0407        DELTA = 1.+.25*(SQLMD+SQMU+SQNU)
0408  C  RODRIGUES MATRIX R
0409        R(1,1)=(1.+.25*(SQLMD-SQMU-SQNU))/DELTA
0410        R(1,2)=(-ANU+.5*ALMDA*AMU)/DELTA
0411        R(1,3)=(AMU+.5*ALMDA*ANU)/DELTA
0412        R(2,1)=(ANU+.5*AMU*ALMDA)/DELTA
0413        R(2,2)=(1.+.25*(-SQLMD+SQMU-SQNU))/DELTA
0414        R(2,3)=(-ALMDA+.5*AMU*ANU)/DELTA
0415        R(3,1)=(-AMU+.5*ANU*ALMDA)/DELTA
0416        R(3,2)=(ALMDA+.5*ANU*AMU)/DELTA
0417        R(3,3)=(1.+.25*(-SQLMD-SQMU+SQNU))/DELTA
0418  C  TRANSLATION VECTOR T
0419        VECT(1) = TEMP(1)
0420        VECT(2) = TEMP(2)
0421        VECT(3) = TEMP(3)
0422        RETURN
0423        END
```

```
0424  C
0425  C  SUBROUTINE NULEG
0426  C  TO COMPUTE NEW LEG LENGTHS GIVEN T AND R
0427  C  AND ALSO TOTAL NUMBER OF MOTOR STEPS
0428  C
0429        SUBROUTINE NULEG(R,VECT,NSTEP)
0430        DOUBLE PRECISION DGLX(3),DGLY(3),DGLZ(3)
0431       1,NGLX,NGLY,NGLZ
0432       1,GENRL(6),A(3,3)
0433        REAL B(3),TOP(3),SQTOP(3),ALEG(6),GAMMA(3)
0434       1,AGLX(3),AGLY(3),AGLZ(3),BGLX(3),BGLY(3),BGLZ(3)
0435       1,ABX(3),ABY(3),ABZ(3)
0436       1,XLEG(6),YLEG(6),ZLEG(6)
0437       1,MOD
0438       1,R(3,3),VECT(3)
0439       1,ESX(3),ESY(3),ESZ(3),DSX(3),DSY(3),DSZ(3)
0440       1,EPX(3),EPY(3),EPZ(3)
0441       1,PINX(3),PINY(3),PINZ(3),ROT(3,3)
0442        INTEGER IWRIT,ERROR
0443       1,NSTEP(6)
0444        COMMON IDEV,IDATA(6),JFAIL,IFLAG
0445        COMMON P(3),Q(3),COSGA(3),SINGA(3),U(3),V(3),C(3),H(3)
0446       1,SQTOP,B,GAMMA,TOP
0447       1,DGLX,DGLY,DGLZ,NGLX,NGLY,NGLZ
0448       1,EPX,EPY,EPZ
0449       1,MOD
0450       1,ALEG
0451       1,GENRL
0452       1,A
0453       1,ERROR
0454       1,IWRIT
0455       1,AGLX,AGLY,AGLZ,BGLX,BGLY,BGLZ,ABX,ABY,ABZ
0456       1,UORGX(6),UORGY(6),UORGZ(6),ALEG0(6)
0457       1,AMOTX(6),AMOTY(6),AMOTZ(6),PITCH
0458  C  COORDINATES OF PLATFORM CORNERS E & OF D
0459        DO 100 I=1,3
0460         ESX(I) = R(1,1)*EPX(I)+R(1,2)*EPY(I)+VECT(1)
0461         ESY(I) = R(2,1)*EPX(I)+R(2,2)*EPY(I)+VECT(2)
0462         ESZ(I) = R(3,1)*EPX(I)+R(3,2)*EPY(I)+VECT(3)
0463         DSX(I) = ESX(I)-V(I)*R(1,3)
0464         DSY(I) = ESY(I)-V(I)*R(2,3)
0465         DSZ(I) = ESZ(I)-V(I)*R(3,3)
0466  100    CONTINUE
0467  C  COMPUTE LEG LENGTHS
0468        DO 200 I=1,3
0469  C  COMPUTE VECTOR W ( TOP AXIS OF UNIVERSAL)
0470         XM = DSX(I)-AGLX(I)
0471         YM = DSY(I)-AGLY(I)
0472         ZM = DSZ(I)-AGLZ(I)
0473         DOTMN= XM*R(1,3)+YM*R(2,3)+ZM*R(3,3)
0474         DOTBN= ABX(I)*R(1,3)+ABY(I)*R(2,3)+ABZ(I)*R(3,3)
0475         IF(DOTBN.EQ.0.) GO TO 10
0476         FACT = DOTMN/DOTBN
0477         TEMPX= XM-FACT*ABX(I)
0478         TEMPY= YM-FACT*ABY(I)
0479         TEMPZ= ZM-FACT*ABZ(I)
0480         ALAMD= 1./SQRT(TEMPX**2+TEMPY**2+TEMPZ**2)
0481         WX   = ALAMD*TEMPX
0482         WY   = ALAMD*TEMPY
0483         WZ   = ALAMD*TEMPZ
0484         GO TO 20
0485  10     WX   = ABX(I)/B(I)
0486         WY   = ABY(I)/B(I)
0487         WZ   = ABZ(I)/B(I)
0488  C  COMPUTE VECTOR U ( BOTTOM AXIS OF UNIVERSAL)
0489  20     DOTMW= XM*WX+YM*WY+ZM*WZ
0490         DOTBW= ABX(I)*WX+ABY(I)*WY+ABZ(I)*WZ
0491         FACT = DOTMW/DOTBW
0492         TEMPX= XM-FACT*ABX(I)
0493         TEMPY= YM-FACT*ABY(I)
```

```
0494          TEMPZ= ZM-FACT*ABZ(I)
0495          ALFA = U(I)/SQRT(TEMPX**2+TEMPY**2+TEMPZ**2)
0496          UX     = ALFA*TEMPX
0497          UY     = ALFA*TEMPY
0498          UZ     = ALFA*TEMPZ
0499    C  COMPUTE VECTORS C AND LEGS
0500          CX     = DSX(I)-UX
0501          CY     = DSY(I)-UY
0502          CZ     = DSZ(I)-UZ
0503          J    = 2*I-1
0504          XLEG(J)= CX-AGLX(I)
0505          YLEG(J)= CY-AGLY(I)
0506          ZLEG(J)= CZ-AGLZ(I)
0507          K      = J+1
0508          XLEG(K)= CX-BGLX(I)
0509          YLEG(K)= CY-BGLY(I)
0510          ZLEG(K)  = CZ-BGLZ(I)
0511          ALEG(J)  = SQRT(XLEG(J)**2+YLEG(J)**2+ZLEG(J)**2)
0512          ALEG(K)  = SQRT(XLEG(K)**2+YLEG(K)**2+ZLEG(K)**2)
0513    C  COMPUTE PIN AXIS
0514    C  PIN(I) = (LEG(K)^LEG(J))/MOD(LEG(K)^LEG(J))
0515          PINX(I)  = YLEG(K)*ZLEG(J)-YLEG(J)*ZLEG(K)
0516          PINY(I) =-XLEG(K)*ZLEG(J)+XLEG(J)*ZLEG(K)
0517          PINZ(I)  = XLEG(K)*YLEG(J)-XLEG(J)*YLEG(K)
0518         PINLE = SQRT(PINX(I)*PINX(I)+PINY(I)*PINY(I)+PINZ(I)*PINZ(I))
0519          PINX(I)  = PINX(I)/PINLE
0520          PINY(I)  = PINY(I)/PINLE
0521          PINZ(I)  = PINZ(I)/PINLE
0522    200   CONTINUE
0523    C
0524    C
0525    C  TOTAL NUMBER OF MOTOR STEPS
0526    C
0527         DO 300 I=1,6
0528          J = (I+1)/2
0529    C  INTERMEDIATE MOTOR UNIVERSAL AXIS,UINT,CORRESPONDING TO ORIGINAL
0530    C  POSITION OF ULEG , WHICH IS THE SAME AS THAT OF PIN
0531    C  UINT =(PIN ^ AMOTO)/MOD(PIN^AMOTO)    AMOTO = MOTOR AXIS
0532          UINTX = PINY(J)*AMOTZ(I)-PINZ(J)*AMOTY(I)
0533          UINTY =-PINX(J)*AMOTZ(I)+PINZ(J)*AMOTX(I)
0534          UINTZ = PINX(J)*AMOTY(I)-PINY(J)*AMOTX(I)
0535         ULENG = SQRT(UINTX**2+UINTY**2+UINTZ**2)
0536         UINTX = UINTX/ULENG
0537         UINTY = UINTY/ULENG
0538         UINTZ = UINTZ/ULENG
0539    C  ANGLE BETWEEN UINT AND ORIGINAL POSITION OF MOTOR UNIVERSAL AXIS
0540    C  POSITIVE : FROM ORIGINAL TO INTERMEDIATE POSITION , ANTICLOCKWISE
0541    C               ABOUT AMOTO
0542          COSAN = UORGX(I)*UINTX+UORGY(I)*UINTY+UORGZ(I)*UINTZ
0543          SINAN =(UORGX(I)*UINTY-UORGY(I)*UINTX)/AMOTZ(I)
0544    C  SMALL ANGLE : -90 DEG < ANGL1 < +90 DEG
0545          ANGL1 = ATAN(SINAN/COSAN)
0546    C  INTEGRAL NUMBER OF REVOLUTIONS AND SCREW ANGLE
0547          DELEG = ALEG(I) - ALEG0(I)
0548           IF(DELEG.LT.0.) ERROR = 5
0549          RMAIN = AMOD(DELEG,PITCH)
0550          NTURN =(DELEG-RMAIN)/PITCH
0551          SCREW = -6.2831852*RMAIN/PITCH
0552    C  IF SREW ANGLE IS LESS THAN 1 DEG , ANGL2 CAN BE IGNORED
0553          IF(ABS(SCREW).GT..0174532) GO TO 250
0554          ANGL2 = 0.
0555          GO TO 290
0556    C  ROTATION MATRIX WITH AXIS LEG(I) AND ROT ANGLE SCREW
0557    250     BL = XLEG(I)/ALEG(I)
0558          BM = YLEG(I)/ALEG(I)
0559          BN = ZLEG(I)/ALEG(I)
0560          BL2 = BL*BL
0561          BM2 = BM*BM
0562          BN2 = BN*BN
0563          COSD= COS(SCREW)
```

```
0564          SIND= SIN(SCREW)
0565          ROT(1,1) = BL2+(1.-BL2)*COSD
0566          ROT(2,1) = BM*BL*(1.-COSD)+BN*SIND
0567          ROT(3,1) = BN*BL*(1.-COSD)-BM*SIND
0568          ROT(1,2) = BL*BM*(1.-COSD)-BN*SIND
0569          ROT(2,2) = BM2+(1.-BM2)*COSD
0570          ROT(3,2) = BN*BM*(1.-COSD)+BL*SIND
0571          ROT(1,3) = BL*BN*(1.-COSD)+BM*SIND
0572          ROT(2,3) = BM*BN*(1.-COSD)-BL*SIND
0573          ROT(3,3) = BN2+(1.-BN2)*COSD
0574    C   LEG'S NEW UNIVERSAL AXIS ( OBTAINED BY ROTATING PIN)
0575          ULEGX = ROT(1,1)*PINX(J)+ROT(1,2)*PINY(J)+ROT(1,3)*PINZ(J)
0576          ULEGY = ROT(2,1)*PINX(J)+ROT(2,2)*PINY(J)+ROT(2,3)*PINZ(J)
0577          ULEGZ = ROT(3,1)*PINX(J)+ROT(3,2)*PINY(J)+ROT(3,3)*PINZ(J)
0578    C   NEW MOTOR UNIVERSAL AXIS
0579    C   UNEW = (ULEG^AMOTO)/MOD(ULEG^AMOTO)
0580          UNEWX = ULEGY*AMOTZ(I)-ULEGZ*AMOTY(I)
0581          UNEWY =-ULEGX*AMOTZ(I)+ULEGZ*AMOTX(I)
0582          UNEWZ = ULEGX*AMOTY(I)-ULEGY*AMOTX(I)
0583          ULENG = SQRT(UNEWX**2+UNEWY**2+UNEWZ**2)
0584          UNEWX = UNEWX/ULENG
0585          UNEWY = UNEWY/ULENG
0586          UNEWZ = UNEWZ/ULENG
0587    C   ANGLE BETWEEN UNEW AND UINT
0588    C   POSITIVE : FROM INT TO NEW POSITION ,ANTICLOCKWISE ABOUT AMOTO
0589          COSAN = UNEWX*UINTX+UNEWY*UINTY+UNEWZ*UINTZ
0590          SINAN = (UINTX*UNEWY-UINTY*UNEWX)/AMOTZ(I)
0591         ANGL2 = ATAN(SINAN/COSAN)
0592    C   ANGL2 IS ALWAYS NEGATIVE : -360 DEG < ANGL2 < 0 DEG
0593          IF((COSAN.EQ.0.).AND.(SINAN.GT.0.)) ANGL2 = -4.7123889
0594         IF((COSAN.EQ.0.).AND.(SINAN.LT.0.)) ANGL2 =-1.5707963
0595          IF(COSAN.LT.0.) ANGL2 = ANGL2-3.1415926
0596          IF(ANGL2.GT.0.) ANGL2 = ANGL2-6.2831852
0597    C   TOTAL ANGLE BETWEEN UNEW AND ORIGINAL POSITION OF MOTOR
0598    C   UNIVERSAL AXIS
0599    C   POSITIVE: FROM TO NEW POSITION , ANTICLOCKWISE ABOUT AMOTO
0600    290     TANGL = ANGL1+ANGL2
0601    C   TOTAL NUMBER OF MOTOR STEPS
0602          NSTEP(I) = (-TANGL*100.)/3.1415926+NTURN*200
0603    300    CONTINUE
0604          IF(IWRIT.EQ.0) GO TO 210
0605          WRITE(6,55)
0606    55      FORMAT(" COMPUTED LEG LENGTHS:")
0607          WRITE(6,77) ALEG
0608          WRITE(6,66)
0609    66      FORMAT(" TOTAL MOTOR STEPS:")
0610          WRITE(6,88) NSTEP
0611    77      FORMAT(3E20.10)
0612    88      FORMAT(6I10)
0613    210    RETURN
0614          END
```

```
0615  C    PHAM D.T.
0616  C    UNIVERSAL TABLE ANALYSIS
0617  C      EQUATE 3 DISTANCES (3 EQUATIONS) AND CONSTRAIN 3 AXES (3 EQUATI
0618  C      ACCEPT 6 LEG LENGTHS ,  COMPUTE 6 GENERALISED COORDINATES
0619  C      21/01/1978
0620         SUBROUTINE CORD4
0621         DOUBLE PRECISION X(6)
0622        1,DGLX(3),DGLY(3),DGLZ(3),NGLX,NGLY,NGLZ
0623        1,SUM,DIFFR,GENRL(6),IPLAT(3),JPLAT(3),KPLAT(3),A(3,3)
0624        1,DELTA,SINDE,AL,AM,AN,DETAN,ALMDA,AMU,ANU
0625        1,TRACE
0626         REAL B(3),TOP(3),SQTOP(3),ALEG(6),GAMMA(3)
0627        1,AGLX(3),AGLY(3),AGLZ(3),BGLX(3),BGLY(3),BGLZ(3)
0628        1,ABX(3),ABY(3),ABZ(3)
0629        1,MOD
0630        1,EPX(3),EPY(3),EPZ(3)
0631         INTEGER IWRIT,ERROR
0632         COMMON IDEV,IDATA(6),JFAIL,IFLAG
0633         COMMON P(3),Q(3),COSGA(3),SINGA(3),U(3),V(3),C(3),H(3)
0634        1,SQTOP,B,GAMMA,TOP
0635        1,DGLX,DGLY,DGLZ,NGLX,NGLY,NGLZ
0636        1,EPX,EPY,EPZ
0637        1,MOD
0638        1,ALEG
0639        1,GENRL
0640        1,A
0641        1,ERROR
0642        1,IWRIT
0643        1,AGLX,AGLY,AGLZ,BGLX,BGLY,BGLZ,ABX,ABY,ABZ
0644        1,UORGX(6),UORGY(6),UORGZ(6),ALEG0(6)
0645        1,AMOTX(6),AMOTY(6),AMOTZ(6),PITCH
0646         EQUIVALENCE (A(1,1),IPLAT),(A(1,2),JPLAT),(A(1,3),KPLAT)
0647  C
0648  C  CHECK DATA
0649         ERROR = 0
0650         DO 100 I=1,6
0651          IF((ALEG(I).LT.300.).OR.(ALEG(I).GT.350.)) GO TO 40
0652  100    CONTINUE
0653         IF(IWRIT.EQ.0)GO TO 10
0654         WRITE(6,11)
0655  11     FORMAT(" INPUT LEG LENGTHS:")
0656         WRITE(6,22)ALEG
0657  22     FORMAT(3E20.10)
0658  C
0659  C CALCULATE PARAMETERS
0660  10     C(1) = .5*(B(1)+(ALEG(1)**2-ALEG(2)**2)/B(1))
0661         C(2) = .5*(B(2)+(ALEG(3)**2-ALEG(4)**2)/B(2))
0662         C(3) = .5*(B(3)+(ALEG(5)**2-ALEG(6)**2)/B(3))
0663         H(1) = SQRT(ALEG(1)**2-C(1)**2)
0664         H(2) = SQRT(ALEG(3)**2-C(2)**2)
0665         H(3) = SQRT(ALEG(5)**2-C(3)**2)
0666         CDUM = .5*(TOP(1)+(TOP(3)**2-TOP(2)**2)/TOP(1))
0667         HDUM = SQRT(TOP(3)**2-CDUM**2)
0668         MOD  = HDUM*TOP(1)
0669         DO 120 I =1,3
0670          COSGA(I) = COS(.01745329252*GAMMA(I))
0671          SINGA(I) = SIN(.01745329252*GAMMA(I))
0672          SQTOP(I) = TOP(I)**2
0673          X(I)     = 1.5707963268
0674          X(I+3)   = 1.5707963268
0675  120    CONTINUE
0676         N = 6
0677  C
0678  C CALL NLNEQ
0679         CALL NLNEQ(X,N,ERROR)
0680         IF(ERROR.NE.0) GO TO 41
0681  C  CALCULATE GENERALISED COORD 1,2,3, OF PLATFORM
0682  C  (COORDINATES OF PLATFORM ORIGIN)
0683         SUM1     = V(1)+V(2)+V(3)
0684         GENRL(1)=(DGLX(1)+DGLX(2)+DGLX(3)+SUM1*NGLX)/3.
```

```
0685          GENRL(2)=(DGLY(1)+DGLY(2)+DGLY(3)+SUM1*NGLY)/3.
0686          GENRL(3)=(DGLZ(1)+DGLZ(2)+DGLZ(3)+SUM1*NGLZ)/3.
0687    C   CALCULATE COORDINATES OF PLATFORM CARTESIAN AXES
0688          KPLAT(1)=NGLX
0689          KPLAT(2)=NGLY
0690          KPLAT(3)=NGLZ
0691          IPLAT(1)=(DGLX(2)-DGLX(3))/TOP(2)
0692          IPLAT(2)=(DGLY(2)-DGLY(3))/TOP(2)
0693          IPLAT(3)=(DGLZ(2)-DGLZ(3))/TOP(2)
0694          JPLAT(1)=KPLAT(2)*IPLAT(3)-KPLAT(3)*IPLAT(2)
0695          JPLAT(2)=KPLAT(3)*IPLAT(1)-KPLAT(1)*IPLAT(3)
0696          JPLAT(3)=KPLAT(1)*IPLAT(2)-KPLAT(2)*IPLAT(1)
0697    C   CALCULATE GENERALISED COORDINATES 4,5,6, OF PLATFORM
0698    C   (RODRIGUES PARAMETERS)
0699    C   NOTE THE DEVIATION FROM THE PAPER BY HMCC AND PDT
0700          TRACE = (A(1,1)+A(2,2)+A(3,3)-1.D0)/2.
0701    C     TEST FOR THE CASE OF SMALL ROTATIONS ABOUT THE
0702    C       ORIGINAL PLATFORM LOCATION
0703          IF((TRACE.GT.1.D0).OR.((DABS(A(1,1)-1.D0).LT.1D-6)
0704         1.AND.(DABS(A(2,2)-1.D0).LT.1.D-6)
0705         1.AND.(DABS(A(3,3)-1.D0).LT.1.D-6))) TRACE=1.D0
0706          DELTA = DATAN((DSQRT(1.D0-TRACE**2)/TRACE))
0707          SINDE=DSIN(DELTA)
0708          IF(SINDE.LT.1.D-10)GO TO 20
0709          AL = (A(3,2)-A(2,3))/(2.*SINDE)
0710          AM = (A(1,3)-A(3,1))/(2.*SINDE)
0711          AN = (A(2,1)-A(1,2))/(2.*SINDE)
0712          DETAN = DSIN(DELTA/2.)/DCOS(DELTA/2.)
0713          ALMDA = 2.*DETAN*AL
0714          AMU   = 2.*DETAN*AM
0715          ANU   = 2.*DETAN*AN
0716          GENRL(4) = ALMDA
0717          GENRL(5) = AMU
0718          GENRL(6) = ANU
0719          GO TO 30
0720    C   TRIVIAL CASE:SINDE=0,IE DELTA = 0 (SINCE WE ASSUME SMALL
0721    C   DELTA
0722    20    GENRL(4) = A(3,2)
0723          GENRL(5) = A(1,3)
0724          GENRL(6) = A(2,1)
0725    30    IF(IWRIT.EQ.0) GO TO 41
0726          WRITE(6,88)
0727    88    FORMAT(" COMPUTED COORDINATES:")
0728          WRITE(6,99)GENRL
0729    99    FORMAT(3E20.6)
0730          GO TO 41
0731    C
0732    C
0733    40    ERROR = 1
0734    41    RETURN
0735          END
```

```
0736  C
0737  C  SUBROUTINE FUNCT
0738        SUBROUTINE FUNCT(X,I,F)
0739        DOUBLE PRECISION X(6)
0740       1,DGLX(3),DGLY(3),DGLZ(3),NGLX,NGLY,NGLZ
0741       1,DX(3),DY(3),DZ(3),COSFI(3),SINFI(3),COSSI(3),SINSI(3)
0742       1,DIFX(3),DIFY(3),DIFZ(3)
0743       1,UGLX(3),UGLY(3),UGLZ(3)
0744       1,F
0745       1,GENRL(6)
0746       1,A(3,3)
0747        REAL SQTOP(3)
0748       1,AGLX(3),AGLY(3),AGLZ(3),BGLX(3),BGLY(3),BGLZ(3)
0749       1,ABX(3),ABY(3),ABZ(3)
0750       1,ALEG(6)
0751       1,B(3),GAMMA(3),TOP(3)
0752       1,MOD
0753       1,EPX(3),EPY(3),EPZ(3)
0754        INTEGER ERROR,IWRIT
0755        COMMON IDEV,IDATA(6),JFAIL,IFLAG
0756        COMMON P(3),Q(3),COSGA(3),SINGA(3),U(3),V(3),C(3),H(3)
0757       1,SQTOP,B,GAMMA,TOP
0758       1,DGLX,DGLY,DGLZ,NGLX,NGLY,NGLZ
0759       1,EPX,EPY,EPZ
0760       1,MOD
0761       1,ALEG
0762       1,GENRL
0763       1,A
0764       1,ERROR
0765       1,IWRIT
0766       1,AGLX,AGLY,AGLZ,BGLX,BGLY,BGLZ,ABX,ABY,ABZ
0767       1,UORGX(6),UORGY(6),UORGZ(6),ALEG0(6)
0768       1,AMOTX(6),AMOTY(6),AMOTZ(6),PITCH
0769  C  CALCULATE COS,SIN OF GUESSED ANGLES
0770  C        PHI(K) = X(K)   , K=1,3
0771  C        PSI(K) = X(K+3) , K=1,3
0772  C  CALCULATE LOCAL AND GLOBAL COORD OF D
0773        DO 100 K=1,3
0774         COSFI(K)= DCOS(X(K))
0775         SINFI(K)= DSIN(X(K))
0776         COSSI(K)= DCOS(X(K+3))
0777         SINSI(K)= DSIN(X(K+3))
0778         DX(K)    =-U(K)*COSSI(K)+C(K)
0779         DY(K)    = U(K)*COSFI(K)*SINSI(K)+H(K)*COSFI(K)
0780         DZ(K)    = U(K)*SINFI(K)*SINSI(K)+H(K)*SINFI(K)
0781         DGLX(K) = COSGA(K)*DX(K)-SINGA(K)*DY(K)+P(K)
0782         DGLY(K) = SINGA(K)*DX(K)+COSGA(K)*DY(K)+Q(K)
0783         DGLZ(K) = DZ(K)
0784  100    CONTINUE
0785  C  CALCULATE D1D2,D2D3,D1D3
0786        DIFX(1) = DGLX(2)-DGLX(1)
0787        DIFY(1) = DGLY(2)-DGLY(1)
0788        DIFZ(1) = DGLZ(2)-DGLZ(1)
0789        DIFX(2) = DGLX(3)-DGLX(2)
0790        DIFY(2) = DGLY(3)-DGLY(2)
0791        DIFZ(2) = DGLZ(3)-DGLZ(2)
0792        DIFX(3) = DGLX(3)-DGLX(1)
0793        DIFY(3) = DGLY(3)-DGLY(1)
0794        DIFZ(3) = DGLZ(3)-DGLZ(1)
0795  C  CALCULATE F , FOR I=1,3 (EQUATING DISTANCE BETWEEN TWO POINTS
0796  C   D AND LENGTH OF PLATFORM TOP )
0797        GO TO (10,20,30,35,35,35) I
0798  10    F = DIFZ(1)**2+DIFY(1)**2+DIFX(1)**2-SQTOP(1)
0799        RETURN
0800  20    F = DIFZ(2)**2+DIFY(2)**2+DIFX(2)**2-SQTOP(2)
0801        RETURN
0802  30    F = DIFZ(3)**2+DIFY(3)**2+DIFX(3)**2-SQTOP(3)
0803        RETURN
0804  C  CALCULATE GLOBAL COORD OF NORMAL TO PLANE D1D2D3
0805  C     NORMAL=(D1D2^D1D3)/MOD(D1D2^D1D3)
```

```
0806   35     NGLX = (DIFY(1)*DIFZ(3)-DIFY(3)*DIFZ(1))/MOD
0807          NGLY = (DIFX(3)*DIFZ(1)-DIFX(1)*DIFZ(3))/MOD
0808          NGLZ = (DIFX(1)*DIFY(3)-DIFX(3)*DIFY(1))/MOD
0809   C  CALCULATE GLOBAL COORDINATES OF SECOND UNIVERSAL AXIS(AXIS I3)
0810          DO 110 K=1,3
0811          UGLX(K) = COSGA(K)*SINSI(K)-SINGA(K)*COSFI(K)*COSSI(K)
0812          UGLY(K) = SINGA(K)*SINSI(K)+COSGA(K)*COSFI(K)*COSSI(K)
0813          UGLZ(K) = SINFI(K)*COSSI(K)
0814   110    CONTINUE
0815   C  CALCULATE F ,FOR I=4,6 (CONSTRAINING SECOND UNIVERSAL AXIS
0816   C     TO LIE IN PLANE D1D2D3)
0817          L = I-3
0818          GO TO (40,50,60) L
0819   40     F = UGLX(1)*NGLX+UGLY(1)*NGLY+UGLZ(1)*NGLZ
0820          RETURN
0821   50     F = UGLX(2)*NGLX+UGLY(2)*NGLY+UGLZ(2)*NGLZ
0822          RETURN
0823   60     F = UGLX(3)*NGLX+UGLY(3)*NGLY+UGLZ(3)*NGLZ
0824          RETURN
0825          END
```

```
0826  C
0827  C
0828  C    SUBROUTINE NLNEQ
0829           SUBROUTINE NLNEQ(X,N,ERROR)
0830          DOUBLE PRECISION X(6),DX,FACTR
0831         1 ,WK(50)
0832         1,TMPDJ(10),XLAST(10)
0833         1 ,F,FPLUS,TEMP,MXDIJ,JIDXM
0834         1 ,EPS,ERS
0835          INTEGER STOP,FREQ,ITNO,I,NZERO,NMINI,FOUND,CONVG,NOIT
0836         1,ERROR
0837         1,ORDER
0838          DIMENSION KEY(10),ORDER(10)
0839          ITNO = 0
0840          FREQ = 0
0841          EPS = 1.D-5
0842          ERS = 5.D-6
0843          PREC= 1.D-6
0844          DELTA = 1.D-7
0845          STOP= 30
0846                    IF (FREQ .EQ. 0) GO TO 5
0847                    WRITE(6,55)ITNO
0848                    WRITE(6,44)X
0849  44        FORMAT(3E20.10)
0850  5        DO 900 NOIT = 1,STOP
0851  C       EXPAND EACH FUNCTION AND SOLVE FOR THE
0852  C       THE LARGEST DERIVATIVE.ELIMINATE THE CORRESPONDING VARIABLE
0853                    FMAX = 0
0854                    NZERO = 0
0855          DO 10 J = 1,N
0856  10                KEY(J) = 1
0857          DO  800 I = 1,N
0858                    NMINI = N - I
0859           FACTR=1.0E-4
0860  C      EVALUATE THE I-TH FUNCTION FOR THE CURRENT X VALUES AND CHECK
0861  C      FOR ZERO
0862                    CALL FUNCT(X,I,F)
0863                    IF (DABS(F).LT.EPS) NZERO=NZERO+1
0864                    IF (NZERO .EQ. N) GO TO 950
0865                    FMAX = DMAX1(FMAX,DABS(F))
0866  C      CALCULATE THE N-I1 PARTIAL DERIVATIVES FOR EACH VARIABLE YET
0867  C      TO BE ELIMINATED
0868                    FOUND = 0
0869  50                 FACTR = FACTR * 10
0870                     IF (FACTR.GT.1.0) GO TO 925
0871                         MXDIJ = DELTA
0872                    MAXJ = 0
0873           DO 100 J = 1,N
0874                     TMPDJ(J) = 0
0875                    IF (KEY(J) .EQ. 0) GO TO 100
0876                    FOUND = FOUND +1
0877                    TEMP = X(J)
0878                     DX = TEMP * FACTR
0879                    DX = DMIN1(FMAX,DX)
0880                    IF (DX .LT. PREC) DX = PREC
0881           X(J) = TEMP +DX
0882                     CALL UPDAX(N,X,I-1,ORDER,WK,KEY)
0883                    CALL FUNCT(X,I,FPLUS)
0884                    X(J) = TEMP
0885                    TEMP = (FPLUS - F)/DX
0886                     TMPDJ(J) = TEMP
0887  C      FIND THE LARGEST DERIVATIVE IN ORDER TO ELIMINATE THE
0888  C      CORRESPONDING VARIABLE
0889                     IF (DABS(MXDIJ).GT.DABS(TEMP)) GO TO 90
0890                     MXDIJ = TEMP
0891                    MAXJ = J
0892  90                IF (FOUND .GT. NMINI) J = N+1
0893  100    CONTINUE
0894                     TEMP = DABS(MXDIJ)
0895                    IF (TEMP .LE. DELTA) GO TO 50
```

```
0896                    JIDXM = 1.0/MXDIJ
0897                    ORDER(I) = MAXJ
0898                   KEY(MAXJ) = 0
0899  C     SAVE THE PARTIAL DERIVATIVES IN ARRAY DIJ .
0900                   WK (I) = -F
0901                   INDX = I * N -(I*(I-1))/2
0902          DO 300 J = 1,N
0903                   IF (KEY(J) .EQ. 0) GO TO 300
0904                   INDX = INDX + 1
0905                    TEMP = TMPDJ(J)
0906                    WK (INDX) = TEMP * JIDXM
0907                   WK (I) = WK (I) + TEMP * X(J)
0908  300     CONTINUE
0909                    WK (I) = WK (I) * JIDXM + X(MAXJ)
0910                    CALL UPDAX(N,X,I,ORDER,WK,KEY)
0911  800     CONTINUE
0912  C     AFTER EACH ITERATION CHECK FOR CONVERGENCE
0913          DO 850 J = 1,N
0914                   TEMP = (XLAST(J)-X(J))/X(J)
0915                   IF (DABS(TEMP) .GT. ERS) GO TO 875
0916  850     CONTINUE
0917                    CONVG = CONVG + 1
0918                    IF (CONVG.GT.3) GO TO 950
0919                   GO TO 900
0920  875              CONVG = 1
0921          DO  880 J = 1,N
0922                   XLAST(J) = X(J)
0923  880     CONTINUE
0924  C     MORE ITERATIONS NEEDED
0925                   ITNO = ITNO + 1
0926                   IF (FREQ .EQ. 0) GO TO 900
0927                 J = MOD(ITNO,FREQ)
0928                   IF (J .NE.  0) GO TO 900
0929          WRITE(6,55) ITNO
0930  55      FORMAT(1X, 13HITERATION NO.,I5)
0931          WRITE(6,44)X
0932  900     CONTINUE
0933  C     MAXIT EXCEDED WITHOUT CONVERGENCE
0934          ERROR = 2
0935                   WRITE(6,11)
0936  11      FORMAT(" MAXIT EXCEDED WITHOUT CONVERGENCE ")
0937                   GO TO 990
0938  C     SINGULAR JACOBIAN
0939  925     ERROR=3
0940          WRITE(6,22)
0941  22      FORMAT(1X,"SINGULAR JACOBIAN. TRY A DIFFERENT INITIAL GUESS")
0942                   GO TO 990
0943  950              WRITE(6,33)  ITNO
0944  33      FORMAT(1X,"CONVERGED IN ",I5," ITERATIONS. ")
0945  990               DO 995 I=1,N
0946                    CALL FUNCT(X,I,TMPDJ(I))
0947  995     CONTINUE
0948  999              STOP = ITNO
0949          RETURN
0950          END
```

```
0951          SUBROUTINE UPDAX(N,X,M,ORDER,DIJ,KEY)
0952         DOUBLE PRECISION X(6),DIJ(50)
0953         INTEGER  J,K,L,MAXJ
0954        1,ORDER
0955         DIMENSION KEY(10),ORDER(10)
0956                   IF (M .LT. 1) GO TO 400
0957                   DO 200 K = 1,M
0958                   L = M - K + 1
0959                   INDX = L * N -(L*(L-1))/2
0960                   TEMP = DIJ(L)
0961                   MAXJ = ORDER(L)
0962         DO  100 J = 1,N
0963                   IF (KEY(J) .EQ. 0) GO TO 100
0964                   INDX = INDX + 1
0965                   TEMP = TEMP -DIJ(INDX) *X(J)
0966   100   CONTINUE
0967                   X(MAXJ) = TEMP
0968                   KEY(MAXJ) = 1
0969   200   CONTINUE
0970         DO 300 J = 1,M
0971   300             KEY(ORDER(J)) = 0
0972         CONTINUE
0973   400   RETURN
0974         END
0975          END$
**** LIST END ****
```

```
0001  ASMB,R
0002  *  PHAM D.T.
0003  *  30/03/1978
0004  * AMENDED VERSION 02/08/1979 : INITL SUBROUTINE NOW INCORPORATES
0005  *                             THE INITIALISATION OF THE LAB
0006  *                             TERMINAL CONTROLLER
0007  *
0008  *  SUBROUTINE OMPLX : OUTPUT MULTIPLEXER
0009  *     ACCEPTS ADDRESS OF A SPECIFIED MULTIPLEXER FROM A COMMON
0010  *     BLOCK AND SENDS A STRING OF DATA ALSO TAKEN FROM THE
0011  *     BLOCK TO THE MULTIPLEXER.
0012  *
0013        NAM OMPLX,7
0014        ENT OMPLX
0015        COM IDEV,IDATA(6),JFAIL,IFLAG,DUMMY(327)
0016  OMPLX NOP
0017        CLF 00       DISABLE INTERRUPT SYSTEM
0018        JSB MCLR     RESET MULTIPLEXER SCAN
0019        LDA 45B
0020        STA COUNT    COUNT = -6
0021  *
0022  *  ENABLE THE SPECIFIED MULTIPLEXER
0023  *
0024        LDA ADD.
0025        OTA 14B
0026        STC 14B,C
0027        LDA IDEV
0028        OTA 13B
0029        STC 13B,C
0030        SFS 13B
0031        JMP *-1
0032        CLC 13B
0033  *
0034  *  SEND DATA TO DATA TERMINAL
0035  *
0036        LDA DAT.
0037        OTA 14B
0038        STC 14B,C       THIS INSTRUCTION WAS OMITTED BEFORE
0039        LDB BUFFA
0040  MORE  LDA B,I
0041        OTA 13B
0042        STC 13B,C    STROBE DATA TERMINAL
0043        SFS 13B     TO ENABLE 1ST SET OF BUFFERS.
0044        JMP *-1
0045        CLC 13B
0046        ISZ B
0047        ISZ COUNT
0048        JMP MORE
0049  *
0050  SEVEN CLA
0051        OTA 13B
0052        STC 13B,C    7TH STROBE ( DUMMY )
0053        SFS 13B
0054        JMP *-1
0055        CLC 13B
0056  *
0057  *  TEST FOR PREMATURE INTERRUPTS
0058  *
0059        STF 00     ENABLE INTERRUPTS
0060        NOP
0061        NOP        ALLOW SYSTEM TO
0062        NOP        'SEE' EXTERNAL INTERRUPTS.
0063        CLF 00     DISABLE INTERRUPT SYSTEM.
0064        LDA IFLAG    READ INTERRUPT FLAG
0065        SZA
0066        JMP ERR01    THERE WAS A PREMATURE INTERRUPT.
0067  *
0068  EIGHT CLA
0069        OTA 13B
0070        STC 13B,C    8TH STROBE
```

```
0071          SFS 13B
0072          JMP *-1
0073          CLC 13B
0074    *
0075    *   TEST FOR A LEGAL INTERRUPT
0076    *
0077          LDA 45B
0078          STA COUNT     SET COUNT = -6
0079    *
0080    MOTRY STF 00
0081          NOP
0082          NOP
0083          NOP
0084          CLF 00
0085          LDA IFLAG
0086          SZA
0087          JMP OCCUR
0088    *
0089          ISZ COUNT       THERE WAS NO INTERRUPT ON
0090          JMP MOTRY         8TH STROBE.TRY AGAIN 6 TIMES.
0091          JMP ERRO2
0092    *
0093    OCCUR CPA IDEV        INTERRUPT OCCURRED!
0094          JMP NINTH       AND IT WAS FROM THE DEVICE!
0095          JMP ERRO3
0096    NINTH CLA
0097          OTA 13B
0098          STC 13B,C       9TH STROBE TO LOAD 2ND BUFFER SET.
0099          SFS 13B
0100          JMP *-1
0101          CLC 13B
0102    *
0103          CLA
0104          STA JFAIL      SUCCESS
0105          JMP RET
0106    *
0107    *   ERRORS
0108    *
0109    ERRO1 LDA 54B
0110          STA JFAIL       JFAIL = 1
0111          JMP RET
0112    ERRO2 LDA 55B
0113          STA JFAIL
0114          JMP RET
0115    ERRO3 LDA 56B
0116          STA JFAIL
0117          JMP RET
0118    *
0119    RET   STF 00
0120          JMP OMPLX,I
0121    *
0122    *   DATA AND CONSTANTS
0123    *
0124    BUFFA DEF IDATA
0125    B     EQU 1
0126    COUNT NOP
0127    ADD.  OCT 46000
0128    DAT.  OCT 30000
0129    *
0130    *
```

```
0131  *    SUBROUTINE MCLR : TO RESET COUNTERS OF MULTIPLEXERS
0132  *                      AND START SCANNING FROM 0
0133  *    NOTE : MCLR MUST BE USED WHEN FLAG 00 IS CLEARED .
0134  *     MCLR SHOULD NOT BE CALLED FROM A FORTRAN PROGRAM .
0135  MCLR   NOP
0136         CLC 13B
0137         LDA ADD01
0138         OTA 14B
0139         STC 14B,C
0140         CLA
0141         OTA 13B
0142         STC 13B,C
0143         SFS 13B
0144         JMP *-1
0145         CLC 13B
0146         JMP MCLR,I
0147  ADD01   OCT 46000
0148  *
```

```
0149  *
0150  *  SUBROUTINE IDENT : TO IDENTIFY INTERRUPTING DEVICE
0151  *        ASSUMPTION : ONLY ADDRESS TERMINAL IS ALLOWED
0152  *                     TO INTERRUPT.
0153  *   IDENT IS ACTIVATED BY A TRAP AT LOCATION 14B
0154  *
0155        ENT IDENT
0156  IDENT NOP
0157        CLF 00
0158        STA TEMPA  STORE A AND B
0159        STB TEMPB  TEMPORARILY.
0160  *
0161        JMP ADDEV   BYPASS THE NEXT SECTION
0162  *                  ( HARDWARE FAULT)
0163  *
0164  *  GET ADDRESS OF INTERRUPT TERMINAL
0165  *
0166  ADTER LIA 14B
0167        AND MASK1
0168        ALF,ALF
0169        RAL         ADJUST INPUT ADDRESS.
0170        CPA ADD02
0171        JMP ADDEV   IT WAS ADDRESS TERMINAL.
0172        CLA
0173        CMA,INA
0174        STA IFLAG   SET IFLAG NEGATIVE.
0175        JMP RETUR
0176  *
0177  *  GET ADDRESS OF INTERRUPTING SUB-DEVICE
0178  *
0179  ADDEV LIA 13B
0180        CMA
0181        AND MASK2
0182        STA IFLAG
0183  *
0184  RETUR LDA TEMPA
0185        LDB TEMPB   RESTORE A & B
0186        STF 00
0187        JMP IDENT,I
0188  *
0189  TEMPA NOP
0190  TEMPB NOP
0191  MASK1 OCT 176
0192  MASK2 OCT 17
0193  ADD02  OCT 46000
```

```
0194  *
0195  *    SUBROUTINE ICLR : TO CLEAR AN INTERRUPT CAUSED BY
0196  *                      THE ROBOT SYSTEM,
0197  *                      TO SEND AN END-OF-JOB SIGNAL TO
0198  *                      FREE THE ADDRESS TERMINAL
0199  *                      TO CLEAR THE SOFTWARE INTERRUPT FLAG.
0200  *    ICLR SHOULD NOT BE CALLED BY OMPLX ( PROBLEM WITH FLAG 00 )
0201        ENT ICLR
0202  ICLR  NOP
0203        CLF 00
0204        CLC 13B
0205  *
0206  *   OUTPUT DUMMY ADDRESS TO CONTROL BOX
0207  *
0208        LDA ADD03
0209        OTA 14B
0210        STC 14B,C
0211        LDA DUM.
0212        OTA 13B
0213        STC 13B,C
0214        SFS 13B
0215        JMP *-1
0216        CLC 13B
0217  *
0218  *   OUTPUT OCTAL 17 TO CONTROL BOX
0219  *
0220        LDA CLEAR
0221        OTA 13B
0222        STC 13B,C
0223        SFS 13B
0224        JMP *-1
0225        CLC 13B
0226  *
0227        CLF 13B
0228        CLF 14B
0229  *
0230        CLA
0231        STA IFLAG
0232        STF 00
0233        JMP ICLR,I
0234  *
0235  ADD03  OCT 46000
0236  DUM.   OCT 10
0237  CLEAR  OCT 17
0238  *
```

```
0239  *
0240  *    SUBROUTINE WPANL : TO WRITE TO PANEL A SINGLE WORD
0241  *       CALLED FROM FORTRAN PROGRAM BY CALL WPANL(IOUT)
0242  *
0243  *
0244         ENT WPANL
0245         EXT .ENTR
0246  PARAM BSS 1
0247  WPANL NOP
0248         JSB .ENTR     PARAMETER PASSING BETWEEN
0249         DEF PARAM     FORTRAN AND ASSEMBLY.
0250         CLF 00
0251         CLC 13B
0252  *
0253  *   ADDRESS PANEL
0254  *
0255         LDA ADD04
0256         OTA 14B
0257         STC 14B,C
0258         LDA PANEL
0259         OTA 13B
0260         STC 13B,C
0261         SFS 13B
0262         JMP *-1
0263         CLC 13B
0264  *
0265  *   SEND DATA TO PANEL
0266  *
0267         LDA DAT04
0268         OTA 14B
0269         LDA PARAM,I
0270         OTA 13B
0271         STC 13B,C
0272         SFS 13B
0273         JMP *-1
0274         CLC 13B
0275  *
0276         STF 00
0277         JMP WPANL,I
0278  *
0279  PANEL OCT 16
0280  ADD04  OCT 46000
0281  DAT04  OCT 30000
```

```
0282   *  SUBROUTINE INITL
0283   *             TO INITIALISE THE LAB TERMINAL CONTROLLER
0284   *             TO LOAD TRAP 13B WITH NOP
0285   *             TO LOAD TRAP 14B WITH JSB 1777B,I
0286   *             TO LOAD TRAP 1777B WITH ADDRESS OF IDENT
0287         ENT INITL
0288         EXT EXEC
0289   INITL NOP
0290   *
0291   *  INITIALISE THE LAB TERMINAL CONTROLLER
0292   *
0293         CLF 00         DISABLE INTERRUPT SYSTEM
0294         CLA
0295         OTA 14B        OUTPUT NULL ADDRESS
0296         STC 14B,C       AND FREE BUS
0297         STC 13B,C       CLEAR DMAT
0298         CLC 13B         FLIP FLOPS
0299         STF 00          RE-ENABLE INTERRUPT SYSTEM
0300   *
0301   *   LOAD TRAP 13B WITH NOP
0302   *
0303         LDA NOOP
0304         LDB P13
0305         JSB EXEC
0306         DEF *+2
0307         DEF RCD19  CALL EXEC USING REQUEST CODE -19
0308   *
0309   *   LOAD TRAP 14B WITH JSB 1777B,I
0310   *
0311         LDA GOTRP
0312         LDB P14
0313         JSB EXEC
0314         DEF *+2
0315         DEF RCD19
0316   *
0317   *   LOAD 1777B WITH ADDRESS OF IDENT
0318   *
0319         LDA ADDID
0320         LDB P1777
0321         JSB EXEC
0322         DEF *+2
0323         DEF RCD19
0324   *
0325         JMP INITL,I
0326   *
0327   NOOP  NOP
0328   P13   OCT 13
0329   RCD19 DEC -19
0330   GOTRP JSB 1777B,I
0331   P14   OCT 14
0332   ADDID DEF IDENT
0333   P1777 OCT 1777
0334         END
0335        $END
**** LIST END ****
```

APPENDIX B

HARDWARE FOR CONTROLLING THE WORK TABLE

The hardware for controlling the work table consists of the following main components:

i) A Hewlett-Packard 2100A mini-computer [94].

ii) Two terminal cards [95].

iii) A local control unit.

These components have been sketched in Fig.7.1.

Fig.B.1 shows the connections between the terminal cards and the control unit. It can be seen that one of the cards, UTC 23, is used by the computer for communicating 4-bit addresses with the control unit. The other card, UTC 14, is used by the computer for communicating 16-bit data with the control unit.

The control unit comprises the following main modules:

1. DIFAD and DIFDA, for converting the Differential outputs of the terminals into the Transistor-Transistor Logic (TTL) inputs of the control unit, and the TTL outputs of the control unit to the Differential inputs of the terminals.

2. DECOD, for decoding the 4-bit addresses sent by the computer into 1 of 16 address signals used to 'address' or 'enable' different modules in the control unit.

3. ENCOD, for interrupting the computer and surrendering 4-bit addresses of different modules in the control unit.

4. READO, for storing 16-bit messages from the computer to the front

panel of the control unit.

5. MULTI 01, for sequentially addressing SPEED 01 - SPEED 06, thus enabling each of these modules, in turn, to accept (16-bit) speed control data from the computer.

6. MULTI 02, for sequentially addressing POSIT 01 - POSIT 06, thus enabling each of these modules, in turn, to accept (13- bit) position control data from the computer.

7. MULTI 03, for sequentially addressing POSIT 01 - POSIT 06, thus enabling each of these modules, in turn, to send (13-bit) position feedback data to the computer.

8. READI, for sending 16-bit messages from the switches on the front panel of the control unit to the computer.

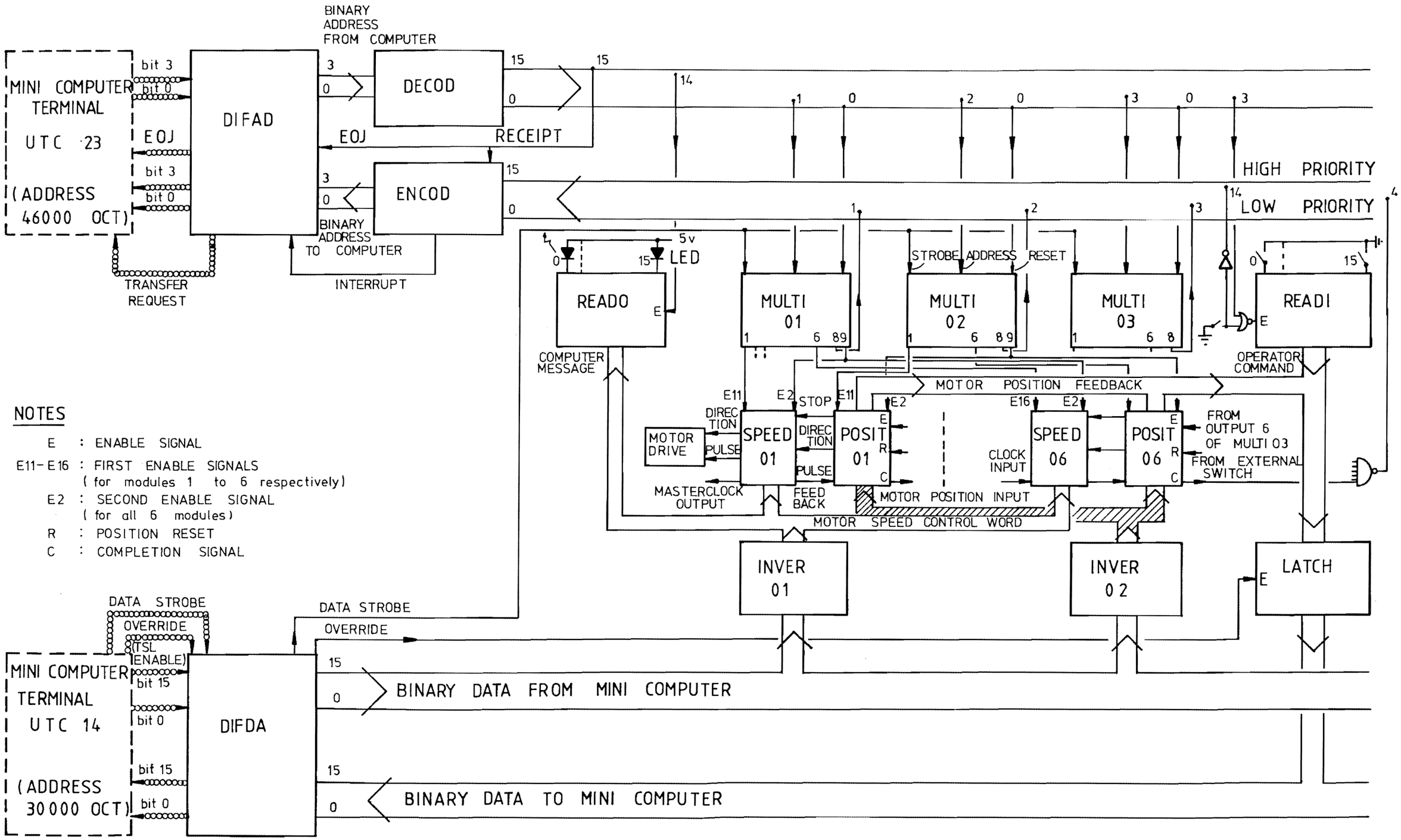
9. SPEED 01 - SPEED 06, for sending pulses to the stepping motor drives (standard commercial units) and controlling the rotational directions of the motors.

10. POSIT 01 - POSIT 06, for inhibiting the pulses from SPEED 01 - SPEED 06 when the motors reach their destinations and controlling the direction signals of SPEED 01 - SPEED 06.

11. INVER 01 and INVER 02, for providing the 'Fan Out' necessary to drive SPEED 01 - SPEED 06 and POSIT 01 - POSIT 06.

12. LATCH, for temporarily storing messages or feedback data to the computer.

The connections between these modules are shown in Fig. B.1. The physical layout of the modules, as viewed from the back-plane of the control unit, is shown in Fig.B.2.



NOTES

E : ENABLE SIGNAL

E11 – E16 : FIRST ENABLE SIGNALS (for modules 1 to 6 respectively)

E2 : SECOND ENABLE SIGNAL (for all 6 modules)

R : POSITION RESET

C : COMPLETION SIGNAL

FIG B.1 BLOCK DIAGRAM OF THE WORK-TABLE CONTROLLER

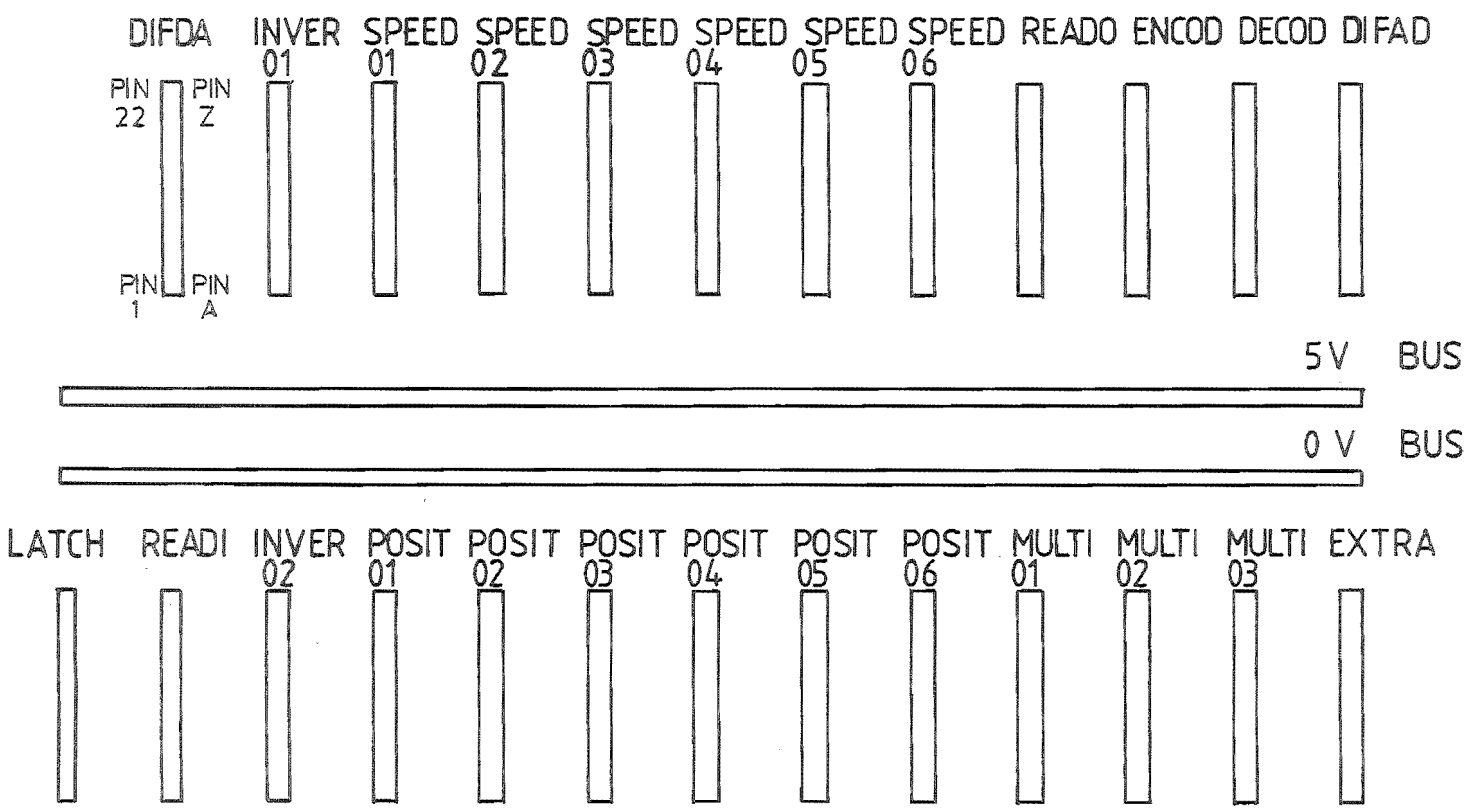


FIG B.2 BACK-PLANE LAYOUT

REFERENCES

1. BOOTHROYD, G. and REDFORD, A.H. Mechanised assembly. McGraw Hill (New York), 1968.

2. TIPPING, W.V. An introduction to mechanical assembly, Business Books, (London), 1969.

3. ASHLEY, J.R. Mechanisation of the assembly process in medium and small batch production. Proc. Roy. Soc. Lond. Vol. A 317, 1970, 455-475.

4. ZOLLMAN, P.M. A multi-station machine for manufacturing small electric lamps. Study 8, Session 3, Second Discussion on Automated Assembling (I Prod E) University of Nottingham, U.K., Sept. 1968.

5. ENCYCLOPAEDIA BRITANNICA Automation, Encyclopaedia Britannica, Vol. 2, 1966, 858-864.

6. PATRICK, B.C. A multi-station in-line transfer machine for assembling a coolant pump. Study 7, Session 3, Second Discussion on Automated Assembling (I Prod E) University of Nottingham, U.K., Sept. 1968.

7. HEGINBOTHAM, W.B. et al. A versatile variable mission assembly machine. Paper A5, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots. University of Nottingham, U.K. March 1976.

8. EISENGREIN, R.H. The future with computer control and/or monitoring. IEEE Trans. on Industrial Electronics and Control Instrumentation, Vol. IECI-21 No. 3, Aug. 1974, 111-115.

9. DUKES, J.M.C. Printed circuits: their design and application. McDonald (London), 1961, 59-65.

10. SEKO, K. and TODA, H. Development and application report in the arc welding and assembly operation by the high-performance robot. Proc. 4th International Symposium on Industrial Robots, Tokyo, Nov. 1974, 487-496.

11. NITZAN, D. and ROSEN, C.A. Programmable industrial automation. IEEE Trans. on Computers, Vol. C25, No. 12, Dec. 1976, 1259-1270.

12. SAVISHCHENKO, V.M. and BESPALOV, V.G. The orientation of components for automatic assembly. Russian Engineering Journal, Vol. 45, No. 5, 1965, 50-52.

13. KARELIN, N.M. and GIREL, A.M. The accurate alignment of parts for automatic assembly. Russian Engineering Journal, Vol. 47, No. 9, 1967, 73-76.

14. WATSON, P. A multidimensional system analysis of the assembly process as performed by a manipulator. SME Technical paper MR76-613, 1976, 16 pp.

15. McCALLION, H. et al. A compliant device for inserting a peg in a hole. The Industrial Robot, Vol. 6, No. 2, June 1979 (to be published).

16. KARELIN, N.M. and GIREL, A.M. Automation of component assembly operations in a rotating magnetic field. Russian Engineering Journal, Vol. 54, No. 10, 1974, 54-57.

17. YAKHIMOVICH, V.A. et al. Automatic assembly of components by the jet method. Russian Engineering Journal, Vol. 50, No. 6, 1970, 58-63.

18. VERITY, P.H. and WALKINGTON, B.R. Pneumatic sensing device, B.E. report, Department of Mechanical Engineering, University of Canterbury, 1974.

19. GORYACHKO, V.I. et al. Vacuum method of automatic assembly, Russian Engineering Journal, Vol. 57, No. 1, 1977, 70-71.

20. ROSEN, C.A. and NITZAN, D. Use of sensors in programmable automation. Computer, Vol. 10, No. 12, December 1977, 12-23.

21. ABRAHAM, R.J. et al. State of the art in adaptable programmable assembly systems. NTIS Report No. NSF/RA 770140, May 1977.

22. PAGE, C.J. et al. Novel techniques for tactile sensing in a three-dimensional environment. Paper C4, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, U.K., March 1976.

23. TAKEYASU, K. et al. Precise insertion control robot and its application. ASME Trans., Series B, Journal of Engineering for Industry, Vol. 98, No. 4, Nov. 1976, 1313-1318.

24. GOTO, T. et al. Compact packaging by robot with tactile sensors. Proc. 2nd International Symposium on Industrial Robots, IIT Research Institute, Chicago, May 1972, 149-159.

25. GOTO, T. et al. Precise insert operation by tactile controlled robot. The Industrial Robot, Vol. 1, No. 5, Sept. 1974, 225-228.

26. D'AURIA, A. and SALMON, M. SIGMA - An integrated general-purpose system for automatic manipulation. Proc. 5th International Symposium on Industrial Robots, IIT Research Institute, Chicago, Sept. 1975, 185-202.

27. PAUL, R. WAVE: A model-based language for manipulator control. The Industrial Robot, Vol. 4, No. 1, March 1977, 10-17.

28. KINOSHITA, G. et al. Pattern classification of the grasped object by the artificial hand. Advance papers of 2nd International Conference on Artificial Intelligence, London, Sept. 1971, 665-669.

29. IVANCEVIK, N.S. Stereometric pattern recognition by artificial touch. Pattern Recognition, Vol. 6, No. 2, October 1974, 77-83.

30. WANG, S.S.M. and WILL, P.M. Sensors for computer controlled mechanical assembly. The Industrial Robot, Vol. 5, No. 1, March 1978, 9-18.

31. KINOSHITA, G. et al. A pattern classification by dynamic tactile sense information processing. Pattern Recognition, Vol. 7, No. 4, December 1975, 243-251.

32. LARCOMBE, M.H.E. Tactile sensors, sonar sensors and parallax sensors for robot applications. Paper C3, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, March 1976.

33. GURFINKEL, V.S. et al. Tactile sensitizing of manipulators, Engineering Cybernetics, Vol. 12, No. 6, Nov. - Dec. 1974, 47-56.

34. HILL, J.W. and SWORD, A.J. Manipulation based on sensor-directed control: an integrated end effector and touch sensing system. Proc. 17th Annual Human Factor Society Convention, Washington, D.C., October 1973.

35. WATSON, P.C. and DRAKE, S.H. Pedestal and wrist force sensors for automatic assembly. Proc. 5th International Symposium on Industrial Robots, IIT Research Institute, Chicago, Sept. 1975, 501-511.

36. SALTER, S.H. Arms and the robot. Paper R23, Proc. 1st Conference on Industrial Robot Technology, University of Nottingham, U.K., March 1973.

37. INOUE, H. Computer-controlled bilateral manipulator. Bull. Japan Society of Mechanical Engineers, Vol. 14, No. 69, March 1971, 199-207.

38. BOLLES, R.C. and PAUL, R. The use of sensory feedback in a programmable assembly system. Stanford Artificial Intelligence Laboratory, Memo AIM-220, Stanford University, October 1973.

39. ABRAHAM, R.G. et al. Requirement analysis and justification of intelligent robots. Proc. 5th International Symposium on Industrial Robots. IIT Research Institute, Chicago, Sept. 1975, 89-111.

40. ROSEN, C.A. and NITZAN, D. Development in programmable automation. Manufacturing Engineering, Vol. 75, No. 3, Sept. 1975, 26-30.

41. NEVINS, J.L. Sensors for industrial automation. McGraw Hill Yearbook of Science and Technology, 1975, 57-70.

42. ALBUS, J.S. and EVANS, J.M. Jr. Robot systems. Scientific American, Vol. 234, No. 2, Feb. 1976, 77-86B.

43. D'AURIA, A. and SALMON, M. Examples of applications of the SIGMA assembly robot, Paper G5, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, U.K. March 1976.

44. NEVINS, J.L. and WHITNEY, D.E. Research on advanced assembly automation. Computer, Vol. 10, No. 12, December 1977, 24-38.

45. NEVINS, J.L. and WHITNEY, D.E. Computer-controlled assembly. Scientific American, Vol. 238, No. 2, Feb. 1978, 62-74.

46. ROSEN, C.A. et al. Machine intelligence research applied to industrial automation. Sixth Report, SRI Project 4391, Stanford Research Institute, Menlo Park, California, Nov.1976.

47. ROSEN, C.A. et al. Machine intelligence research applied to industrial automation. Seventh Report, SRI Project 4391, Stanford Research Institute, Menlo Park, California, August 1977.

48. McCALLION, H. and WONG, P.C. Some thoughts on the automatic assembly of a peg and a hole. Proc. 4th World Congress on the Theory of Machines and Mechanisms, I. Mech. E., London, Sept. 1975, 347-352.

49. WOHLERT-JENSEN, C.H.H. Techniques for automatic assembly. Ph.D. Thesis, University of Canterbury, 1978.

50. INOUE, H. Force feedback in precise assembly tasks. Memo No. 308 MIT Artificial Intelligence Laboratory, Cambridge, Massachusetts, August 1974.

51. HANAFUSA, H. and ASADA, H. An adaptive control of robot hand equipped with pneumatic proximity sensors, Paper D4, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, U.K., March 1976.

52. MOYLAN, M.J. Fluid logic in simple terms. The Machinery Publishing Co. (London), 2nd Edition, 1971, 156-157.

53. ANON. Proximity switches. Machine Design, Vol. 45, No. 10, 26 April 1973, 33-37.

54. CHANG, K.N. A relative position sensing ultrasonic system for remote position control. M.E. Thesis, University of Canterbury, 1976.

55. UEDA, M. et al. Sensors and systems of an industrial robot. Memoirs of the Faculty of Engineering, Nagoya University, Japan, Vol. 27, No. 2, 1975, 163-207.

56. NITZAN, D. et al. The measurement and use of registered reflectance and range data in scene analysis. IEEE Proceedings, Vol. 65, No. 2, February 1977, 206-220.

57. BUCKLEY, S. and STELSON, K. Phase monitoring for automated inspection, positioning and assembly. SME paper No. AD 770 730, Nov. 1977, 18 pp.

58. WILLIS, W.P. A short-range ultrasonic position detector for automation. M.E. Thesis, University of Canterbury, Feb. 1969.

59. RETICON CORPORATION. Solid state image sensors and systems: Product summary, Reticon Corp., Sunnyvale, California, 1976.

60. ROSEN, C.A. et al. Exploratory research in advanced automation. First Report, SRI Project 2591, Stanford Research Institute, Menlo Park, California, December 1973.

61. CHIEN, R.T. and SNYDER, W.E. Hardware for visual image processing, IEEE Trans. on Circuits and Systems, Vol. CAS-22 No. 6, June 1975, 541-551.

62. McCARTHY, J. A computer with hands, eyes, and ears. AFIPS Conference Proceedings, Fall Joint Computer Conference, Vol. 33, Part One, 1968, 329-338.

63. HEGINBOTHAM, W.B. et al. The Nottingham SIRCH assembly robot, Paper R9, Proc. 1st Conference on Industrial Robot technology, University of Nottingham, U.K. March 1973.

64. HEGINBOTHAM, W.B. et al. A practical visually interactive robot handling system. The Industrial Robot, Vol. 2, No. 2, June 1975, 61-66.

65. TSUBOI, Y. and INOUE, T. Robot assembly system using TV camera. Paper B3, Proc. 3rd Conference on Industrial Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, U.K., March 1976.

66. OLSZTYN, J.T. et al. An application of computer vision to a simulated assembly task. Proc. 1st International Joint Conference on Pattern Recognition, Washington, Oct. 1973, 505-513.

67. EJIRI, M. et al. A prototype intelligent robot that assembles objects from plan drawings. IEEE Trans. on Computers, Vol. C-21, No. 2, February 1972, 161-170.

68. ANON. Automatic bolt tightening and loosening system with visual pattern recognition. Hitachi Review, Vol. 23, No. 4, April 1974, 178-179.

69. AMBLER, A.P. et al. A versatile system for computer-controlled assembly. Artificial Intelligence, Vol. 6, No. 2, Summer 1975, 129-156.

70. WILL, P.M. and GROSSMAN, D.D. An experimental system for computer controlled mechanical assembly. IEEE Transactions on Computers, Vol. C-24, No. 9, Sept. 1975, 879-888.

71. GROSSMAN, D.D. and BLASGEN, M.W. Orienting mechanical parts by computer-controlled manipulator. IBM Research Report RC 4995, IBM T.J. Watson Research Centre, New York. Aug. 1974.

72. WANG, S.S.M. Theoretical study of parts orientation in a manipulator's hand. IEEE Milwaukee Symposium on Automatic Control (and Autonomous Computing), March 1974, Milwaukee, Wisconsin, 226-234b.

73. TAKEYASU, K. et al. An approach to the integrated intelligent robot with multiple sensory feedback: construction and control functions. Proc. 7th International Symposium on Industrial Robots, Tokyo, Oct. 1977, 523-530.

74. KASHIOKA, S. et al. An approach to the integrated intelligent robot with multiple sensory feedback: visual recognition techniques. Proc. 7th International Symposium on Industrial Robots, Tokyo, Oct. 1977, 531-538.

75. DUDA, R.O. and HART, P.E. Pattern classification and scene analysis. Wiley (New York), 1973, Chapter 10, 379-404.

76. WEATHERBURN, C.E. Elementary vector analysis with application to geometry and mechanics. G. Bell (London), 1960, 162-165.

77. BROWN, K.M. A quadratically convergent Newton-like method based upon Gaussian elimination. SIAM Journal of Numerical Analysis, Vol. 6, No.4, Dec.1969, 560 - 569.

78. McCALLION, H. and PHAM, D.T. On measuring errors in a placement task. The Industrial Robot, Vol. 4., No.2, June 1977, 86 - 92.

79. McENTIRE, R.H. Three-dimension accuracy measurement methods for robots. The Industrial Robot, Vol.3, No.3, Sept.1976, 105-112.

80. RODRIGUES, O. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. Journal de Mathématiques Pures et Appliquées, Vol.5, 1st series, December 1840, 380-440.

81. COE, C.J. Theoretical mechanics : a vectorial treatment. McMillan (New York), 1938, Chapter 5, 151-205.

82. THOMPSON, E.H. An introduction to the algebra of matrices with some applications, Adam Hilger (London), 1969, Chapter 8, 124-158.

83. HUNT, K.H. Mechanisms and motion, English University Press (London) 1959, Chapter 7, 73-88.

84. BAGCI, C. Degrees of freedom of motion in mechanisms. ASME Trans. Series B, J. of Engineering for Industry, Vol.93, No.1, Feb.1971, 140-148.

85. DIZIOGIU, B. Theory and practice of spatial mechanisms with special positions of the axes. Mechanism and Machine Theory, Vol.13, No.2, 1978, 139-153.

86. GOUGH, V.E. and WHITEHALL, S.G. Universal tyre test machine. Proc. 9th International Automobile Technical Congress, FISITA, May 1962, I.Mech.E.(London), 117-137.

87. STEWART, D. A platform with six degrees of freedom. Proc. I.Mech.E. (London) 1965-1966, Vol.180. Pt.1, No.15.

88. KOOGLE, T.A. et al. A motion transducer for use in the intact in-vitro human lumbar spine. ASME Trans. J. of Biomechanical Engineering, Vol.99, Ser.K, No.3, Aug. 1977, 160-165.

89. WONG, P.C. Notes on the 6-dof table (unpublished), Department of Mechanical Engineering, University of Canterbury, 1975.

90. WALDRON, K.J. The constraint analysis of mechanisms. Journal of Mechanisms, Vol.1, No.2, 1966, 101-114.

91. VOINEA, R, and A ANASIU, M. Contribution ā l'étude de la structure des chaînes cinématiques. Bul. Inst. Politechnic Bucuresti, Vol.22, No.2, 1960, 29-77.

92. BROWN, K.M. and GEARHART, W.B. Deflation techniques for the calculation of further solutions of a non-linear system. Num. Math (Germany), Vol.16, No.4, 1971, 334-342.

93. McCALLION, H. Vibration of linear mechanical systems. Longman (London), 1973, 135-137.

94. HEWLETT-PACKARD COMPANY. A pocket guide to the 2100 Computer. Hewlett Packard Company (California), 1972.

95. HODGE, J.G. Data acquisition from a range of transducers using a mini-computer based system. M.E. Thesis, University of Canterbury, 1979.